In this term project, you are asked to design a small database system, create and populate this database by using MS Access, Personal ORACLE, or MYSQL, and write a number of application programs and GUls to access the database. The topic of the project is to design, develop, and test the information system for GetAwayCheap.com a new travel set that is being set up with the objective of providing the best services to its customers by keeping their services at the leading edge of technology. The project is in three parts: conceptual design and the requirements analysis (Phase I), database design requirements (Phase II), and application design and development (Phase III).

Important Dates:
- September 17, 2003, Finalize Groups for the Project
- Project Phase I: Monday October 6
- Project Phase II: Monday October 20
- Project Phase III: Monday, December 1 (or Wednesday, December 3) with Demos on Thursday and Friday, December 4 and 5.

PROJECT GUIDELINES

1. This project is a team project. The group size will be 3 members. Each group should do the project independently, though limited discussion among groups is encouraged. For example, asking help with installing a database product, using JDBC to interact with your database, configuring a piece of software, are all valid discussion. However, cooperative design and development among groups is prohibited. All material submitted that represents work copied from elsewhere MUST be explicitly acknowledged.

2. The project will be conducted in three consecutive phases:
   - Phase 1: Requirement Analysis and Specification. The main task of this phase is to perform requirements discovery and analysis using UML diagrams (namely, use-case, state-chart, and activity) for a functional analysis of GetAwayCheap.com and data and information analysis in support of conceptual modeling using the ER design model (required) and business process modeling (as needed). Both of these tasks can be accomplished via the UML tool Together CC, which has all of these diagrams, and under the "Print" option, will generate a printed copy of all of your documentation.
   - Phase 2: Database Design. The main task of this phase is to map the ER model that you have designed to the relational data model and to normalize the relational schemas into either third normal form or BCNF.
   - Phase 3: Application Design, Development, and Testing. One of the main tasks in this phase is to design and write a set of application programs that implement all the functions/processes specified in Phase I. In order to test and run your application program interfaces (APIs), you will need to create the relational schemas for the database, design the data sets that you use for the testing and demo, populate the database with the data sets, and then conduct your testing for each API. In a sense, these two tasks are disjoint: the database can be designed, implemented and tested, independent of the APIs; and, the APIs can be implemented independent of the GUls, tested, interface to the working database, and then integrated with web-based or Java applications. This supports a parallel design and development process among team members during the second half of the semester.
3. The application design and the implementation of your GetAwayCheap.com system must be implemented in Java using either MS Access, MYSQL, Personal Oracle (or Educational Oracle 9i at the School of Engineering Learning Center) as the database management system. If you would like to use other DBMS product instead of the three listed, you must discuss your alternative choice with your instructor and obtain the permission in advance. For all DBMS, you will be responsible for setting up your system in the lab or your own laptop for project demo at the end of the semester, or in the case of Oracle 9i, having your demo in the Learning Center. Note: It may also be possible to design and develop web-based user interfaces that utilize the APIs (via html, applets, and or ASP/JSP) for a subset of the user applications. This issue will be discussed later in the semester.

4. The project description for GetAwayCheap.com is rather generic on purpose, in the sense that it has not been fully specified. One key objective in all three Phases of the project is to extend the project score. Thus, you are encouraged to expand the description with additional functionality and capabilities as warranted to keep three of you busy for the second half of the semester in development.

5. You are required to demonstrate your programs to the instructor. The demonstration will be scheduled in two weekdays: December 4 (1:00pm - 5:00pm) and December 5 (1:00pm - 5:00pm). You should treat these demonstrations as if you were giving them to your customer, so prepare them professionally. It may be useful for your demonstration to contain a short (10-15 minute) PowerPoint presentation in addition to the actual programs (GUIs), APIs, and database that you have designed and have been able to get working. All team members are required to participate in the final demonstration.

6. You are required to submit a word processed project report (Word or PDF) for grading at the end of each phase. You must submit both electronic and hard copies of your report for each phase. For each of the three reports, you must identify the major design decisions that you faced and the design decisions that you made with justifications for those decisions.

7. There are two documents on the web page on individual contribution and final self assessment that are required submissions with the final project. Both should be reviewed and considered at the earliest stages of the semester project. In particular, the individual contribution requires each student to keep a diary of his/her activities, so that I can track that each team members is contributing and making progress. The individual contribution MUST be turned in with each Phase of the project - and is a document that is created and appended throughout the semester (like a diary).

8. Grading policy, subject to adjustment, is as follows:
   - Phase I design report: 20-30%
   - Phase II design report: 20-30%
   - Phase III application programs: 40-60%

   Last semester, the actual percentages were: 25, 25, and 50 for Phases I, II, and III, respectively.
INTRODUCTION

CheapGetAway.com is intended to integrate and unify all activities related to traveling. It is intended for travel agents that are seeking to provide services to their customers, and would also be available electronically for home and business use via both on-line services and network browsers. A preliminary requirements analysis has been conducted that has identified a number of characteristics and features the operations (business processes) and goals of GetAwayCheap.com:

- User interfaces for travel agents, home users, and businesses to provide access to all of the services supported by CheapGetAway.com.
- Sophisticated traveler profiles (TPs) that contain up-to-date information on each traveler and are utilized to provide customized services based on individual preferences.
- An industry-wide hotel/resort component that maintains a detailed database of all lodgings (world-wide) along with the ability to automatically interface with their reservation systems via electronic access.
- Reservation support for hotels, rental car companies, airlines, rail companies, bus companies, cruise ships, etc., through links to their official sites.
- Interactions with existing payment services companies (Visa, MC, AMEX, etc.).

Clearly, within these capabilities there is the potential for significant interactions between new and well-established software systems.

The purpose of CheapGetAway.com is to utilize computing technology in a positive way to enhance, facilitate, and promote this activity. To do so, there is a requirement for interactions with the following individuals:

- Travel agents whose job responsibilities are directly tied to the capabilities and services offered by CheapGetAway.com.
- Home users who access CheapGetAway.com to either make reservations themselves or to check on reservations made through travel agents. Home users should be given the ability to create and update their traveler profiles.
- Business users would have similar access and interactions to home users, with the exception that payment options might also include direct billing of the company by the service or goods provider.
- Hotel/Resort personnel to allow them to update their entries in the industry-wide database.
- Marketing personnel who work for companies that are involved in the travel/vacation industry. These individuals are interested in using non-confidential information in traveler profiles to conduct market surveys, solicit new business, etc. These same companies may also contact travel agents with various offers that are intended to have the agents promote their services (e.g., vacation packages).
- Airlines personnel who access the traveler profiles of the individuals on specific flights to better serve their needs. For example, the TPs may contain dietary restrictions. Similar access will also occur by other transportation related companies (e.g., bus, rail, cruise, etc.)
Government personnel working for the embassies and state departments who access TPs for those individuals making trips to foreign countries.

Security personnel who utilize travel profiles to insure the safety of individuals as they travel. Some of the previous individuals will also need to support interactions with existing commercial software, e.g., to process credit-card payments, etc.

To support the functional and operational requirements of CheapGetAway.com, there must are a number of databases that must be present:

- **Travel Agent Database**: Contains information on the different travel agents that use CheapGetAway.com. Travel agents must register with CheapGetAway.com when they first begin using the system. Information in this database would include name, address, phone numbers, and electronic access (if available).

- **Traveler Profile Database**: Contains both confidential and public information on travelers. Provides a centralized resource for tracking frequent flier miles and other travel bonuses offered by the various parts of the travel and vacation industry. The information contained in this database ranges from the traditional (e.g., name, address, phone, etc.) to more sophisticated information (e.g., passport no., dietary restrictions, medical conditions, etc.).

- **Lodging Database**: The information is this database is very extensive, since it must capture a wide variety of lodging possibilities. Everything from major hotels to bed-and-breakfast inns, from casinos to Disneyland, are possible. In addition to normal information on location, contact information, and rates, it is also reasonable for URLs to be stored to allow automatic access (if desired) to the relevant WWW pages.

- **Rental Car Database**: This contains information on various car-rental companies, transportation alternatives (models and makes - from cars to vans), rates, availability, restrictions, links to rental-car companies web sites, link to car manufacturer web sites (to show makes and models), insurance minimums, limits, and liabilities, and so on.

- **External Databases** which provide access to data repositories that are maintained by external Reservation Systems and Payment Services Companies.

These database are critical for supporting and promoting the sharing and exchange of information among the many individuals that require access to CheapGetAway.com.

All of the previous individuals and services will require specialized user interfaces to handle the information and processing that is required. Some possible interfaces include:

- **Travel Agent Interface**: This interface is critical since it is intended to support a major sector of the travel industry. Since jobs and profits depend on this interface, its reliability is critical to insure maximum availability 24 hours a day. Note that CheapGetAway.com is international, so it must be available at all times. While user friendliness is important, it is likely that travel agent personnel would be trained to use CheapGetAway.com effectively.

- **Home/Business User Interface**: This may be one interface, or two very similar interfaces (i.e., program family) that is accessible via either an on-line service or WWW browser. User friendliness is critical in this interface.

- **Traveler Profile Interface**: This interface is for the different individuals who seek to access information in the TPs. A key issue for this interface is security, since the individuals using it
should only be allowed to access those portions of the TP that have been authorized to their use. For example, credit card information should only be available to the relevant credit card company.

- Hotel/Lodging Interface: This interface allows its users to create/update their entry in the industry-wide database that is maintained.

- Rental Car Interface: This interface allows rental car reservations and linkages to the web pages of rental car companies.

Remember, this is not an exhaustive list of all user interfaces for CheapGetAway.com. However, it is intended to get you started in the process of discerning both the functional and data requirements for the semester project.

PHASE I -- CONCEPTUAL DESIGN REQUIREMENTS

Conceptual design is a combination of requirements analysis, software engineering, and database design, which can be utilized to arrive at a functional and data driven characterization for GetAwayCheap.com, a new travel service that is being set up with the objective of providing the best, and most cost effective, services to its customers by keeping their services at the leading edge of technology. This is clearly a norm for many of the various e-business web sites. You have been hired to set-up an information system and associated user interfaces to assist GetAwayCheap.com in achieving its goals. In Phase I, each group, with members playing the roles of systems analysts/designers/engineers, are required to add, expand, and refine these requirements in order to achieve a richer design. To accomplish this, you are to utilize the preliminary requirements (see INTRODUCTION section) as a basis to arrive at a finalized set of both functional and database requirements.

For overall requirements, you must utilize UML, specifically, use-case, activity, and state-chart diagrams, as supported by Together CC. For database requirements, ER (required) and business process (as needed) diagrams can be utilized - also via Together. The report must include a description of the purpose of this project, and the purpose of this phase of the project. It must describe the problems encountered in this phase, and justify the solution. It must contain all the documentation produced in this phase, including the development of (a) an entity relationship model for the conceptual design, (b) detailed documentation of database content by describing each entity and each relationship, and (c) associated UML diagrams for functional requirements. If you have extended the requirements or refined the requirements presented, you must carefully document and explain these additions/changes in prose. You can use this MS Word document as a basis for a revised, written, prose specification. Note that it may be helpful to read the entire Phase II and Phase III parts of this project specification, to obtain an overall view of the system features and characteristics.

To provide a basis for you to begin to consider both the database and functional requirements, the next section of this specification describes the RenToGo.com child of GetAwayCheap.com.

**RenToGo.com**

In support of automobile rentals, GetAwayCheap.com will have a child web site, RenToGo.com, that will be an information management system that supports a subset of the services involved in an online car rental service (e.g., Alamo.com, Hertz.com). For rental cars, there are three types of users, namely, customer, clerk on duty, and system administrator. Customer is any user willing to rent a car. Clerk on duty represents the employee responsible for checking out and checking in a
CSE2550 Introduction to Databases - Fall 2003 Semester Project
Phases I and II

rental. System administrator is responsible for operating the system. In the rest of this document, we first describe the system interface for each customer type. Then, we present aspects of the system as a whole. Throughout the document, we assume that GetAwayCheap.com can direct you to RenToGo.com with an entry point as shown in Figure 1. Note that these are mockups and can be changed and customized throughout the semester in support of your project goals and objectives.

The Customer

The customer enters the system by selecting Car Rental Service option in the Welcome screen (Figure 1) and clicking ENTER. RenToGo.com then pops up the screen in Figure 2.

Rates and Reservation

If the user clicks on Rates and Reservation button, the system will display the screen in Figure 3, which explains the three simple steps for checking on car rates and making a reservation.

As shown in Figure 4, for the first step in the car reservation process, the user must indicate when and where she will pick and return the car. For this project, we consider that there are only five locations: Boston, Springfield, Hartford, Providence, and Manchester (NH). Each location must keep information of which cars it has (or it will have) available for rental. However, we will assume

Figure 1: Welcome Screen

Figure 2: Customer Main Menu screen

Figure 3: Simple explanation for checking on car rates and making reservation
that the database is centrally managed from one location, say Boston. Based on this information, the system is able to provide the rental availability information in Step 2.

Figure 4: Step 1: Pickup and drop-off location.

In Step 2 (Figure 5), the user selects the car that she wants to rent from a list of available ones. For each car option, the system presents the total cost and the respective daily rate. For this project, you don’t need to worry with the sales tax. The total cost is calculated by just multiplying the number of days by the respective daily rate. The daily rate will be $$.cc in dollars and cents.

In Step 3, the system presents a summary of the rental reservation to the customer. If the customer agrees with the summary, she then clicks on “Complete Reservation” button. Alternatively, she can go back to Step 2 or decide to cancel the reservation at once. We observe that up to this point, no modification has been made to the underlying database. Reservations are only committed when the
customer clicks on the “Complete Reservation” button.

If the customer decides to complete the reservation, the system will pop up the screen in Figure 7. For this project, if a customer wants to make a reservation, she will need to register to RenToGo.com first. In case she has already registered to the system, she needs just to enter her email and password and then click the “Login” button. As we discuss next, the system identifies each registered customer by his or her e-mail address.

If the user is a new customer, when she clicks on the “Register” button, the system shows the screen depicted in Figure 8.

The system uses the e-mail address of the customer to uniquely identify her. We will assume that the e-mail address is a valid one. Therefore, there should not exist any duplication. But the system should still check whether the e-mail address provided by the user has been used before. If this is the case, an error message is displayed, and the customer is requested to enter another e-mail address. The password can be anything that the customer provides. When the customer comes for picking up the car at the selected location, the clerk-on-duty will ask her for a credit card, which will be charged when the car is returned. The customer may decide to provide information on her credit card at pick-up time or at registration time. In case she decides for providing the credit card
information at registration time, she clicks the “File Credit Card Information” button in Figure 8, and the system pops up the screen in Figure 9. The customer can file as many credit cards as she wants. Credit card information can then be deleted or updated using the “Update Profile” option shown in Customer Main Menu screen.

To file a credit card, the customer enters the requested information and clicks “Add Card” button. The system then includes the credit card data and display again the screen shown in Figure 9, this time with the fields cleared. Once the customer has finished entering the credit card information, she clicks “Done” and the screen in Figure 8 is displayed again. The system needs to guarantee that the customer will not enter more than once the same credit card information. If the customer decides to cancel her registration, no reservation will be made, and the Customer Main Menu screen is displayed. However, if the customer clicks the “Register” button in Figure 8, the system will display the screen depicted in Figure 10 and complete the rental reservation.

![Figure 10: Reservation Completed screen.](image)

Before displaying Reservation Completed screen, the system updates the underlying database with the reservation. Note the system will only change the database when the reservation is completed.

**Location-Car Current Availability**

This option presents a snapshot of the current available cars of each type at each location. Using this option, the user can have a general idea of what is available at RenToGo.com at a given moment.

**Manage (Customer) Profile**

When the customer selects the “Edit Profile (Registered Customers Only)” option from the Customer Main Menu screen, the system pops up the following screen:

![Figure 11: Customer Login screen.](image)
Upon successful login, the customer will be presented with the screen shown in Figure 12, from which she will be able to update her first and last name, manage her credit card on file, and check on her rental history. The customer cannot update her e-mail because RenToGo.com uses it as a mechanism for uniquely identifying her within the system.

![Manage Customer Profile screen.](image)

If the customer clicks the “Done” button, the system first commits the changes to her name and driver’s license (if any) and then shows the Customer Main Menu screen. If the customer clicks on “Manage Credit Cards on File”, the system pops up the screen depicted in Figure 13. From this screen, she can browse through all her credit cards on file and update or delete any of them. If she clicks the “Update” button, the system updates the credit card with the new information and remains on the same screen. If the customer clicks the “Delete” button, the respective credit card is removed from the database, and the next filed credit card (if any) is displayed. Finally, if the customer clicks on the “New” button, the system clears all the fields in the screen so she can enter the information for the new credit card. The new credit card will be stored in the database when the customer clicks the “Update” button. If the customer clicks on the “Done” button, the system shows the Manage Customer Profile screen again.

![Manage Credit Cards on File screen.](image)

If the customer clicks on “View Rental History” button on Manage Customer Profile screen, the system pops up the screen shown in Figure 14, which shows the customer rental history in the last month. The status of a rental is one of the following:
- Pending: the customer has completed the reservation but hasn’t picked up the car yet.
- In-process: the customer is using the car.
- Completed: the customer has returned the car.

Observe that a customer may have rented a car from RenToGo.com more than once in the last month. The View Rental History screen should display all rentals (note the scroll bar on the right of the rental history in Figure 14). When the customer clicks the “Done” button in screen of Figure 14, the system shows the Manage Customer Profile again.

The Clerk on Duty

Up to this point, we have described how the customer makes a rental reservation and manage her profile stored in RenToGo.com. In this section, we describe how RenToGo.com handles the car pickup and drop-off processes. We refer to the clerk on duty as the person (user) responsible to taking care of these processes.

Pick-up Rental

When the customer arrives at the location for picking up the rental, the clerk on duty, who from now on we refer as the clerk, logs in to the system by selecting “Clerk on Duty” in the Welcome screen and providing his login and password information in the Login screen shown in Figure 15. Upon successful login, the system pops up the Clerk on Duty Main Menu screen in Figure 16.

![Figure 14: View Rental History screen.](image)

![Figure 15: Clerk on Duty screen.](image)
The rental pickup process consists of two main steps:

Step 1: Get customer e-mail address and rental number (Figure 17).

Step 2: Get customer credit card information to be used for this rental (Figure 18).

Step 1 is straightforward. In Step 2, the clerk asks for the customer to provide the credit card number to be used with this rental. If the credit card has been filed before, then the system pops up the file rental contract shown in Figure 19. Before handing the contract to the customer, the clerk asks the customer for her driver’s license to do a final check. If everything looks ok, the customer signs the contract, and the clerk gives her the car keys and directions to where the car is parked in the parking lot. If in Step 2 the system indicates that the credit card number does not exist in the customer profile, the clerk asks the customer if she wants to provide another credit card number, or if she would like to file this new number in her profile. In case she decides to file the new number, the clerk enters the respective credit information through a screen similar to the one shown in Figure 9.
Drop-off Rental

When the customer arrives at a location for dropping of the car, the clerk clicks on the “Rental Drop-off” button in Clerk on Duty Main Menu screen and follows two simple steps: Step 1: Get customer e-mail address and rental number (similar to screen in Figure 17) Step 2: Print the final receipt for the rental (Figure 20) Before printing out the final receipt, the system needs to check whether the following penalty fees should be applied: Late fee: The customer returns the car on a later day; or, One-way rental fee: The customer returns the car at a different location. The Late fee is calculated by multiplying the number of late days by $2.50. One-way rental fee is $10.00. The system determines whether the drop-off is late by comparing the information on the rental contract and the system current date. After calculating the final rental cost, the customer credit card is charged and the clerk gives the customer the final receipt (Figure 20).
When the clerk finishes the pickup and/or drop-off process, he logs out by clicking on the “Logout” button in the Clerk on Duty Main Menu screen (Figure 16).

**The System Administrator**

The third type of RenToGo.com user is the system administrator, who is responsible for operating the online car rental system. The screen on Figure 21 lists all the administrator tasks. The administrator enters the system by selecting “System Administrator” in the Welcome screen and providing his login information. Login information for the administrator is determined by the DBA.

![Figure 21: Administrator Main Menu screen.](image)

**Manage Employees Information**

When the administrator clicks on “Manage Employees Information” button shown in Figure 21, the system pops up the screen in Figure 22. Using this screen, the administrator can either enter the ID of an existing employee or request for creating a new one. If the administrator enters the employee ID and clicks the “Search” button, the system will first check whether there exists an employee with the provided ID. If there is, then the system displays the associated information as shown in the screen of Figure 23. If there is no employee with the provided ID, the system displays a message indicating that the ID is invalid and pops up again the screen in Figure 22. If the administrator wants to insert a new employee into the system, he clicks on the “New Employee” button. The system then automatically generates the ID for the new employee and displays the screen in Figure 22 with the blank fields. Once the administrator has entered the information of the new employee, he clicks the “Update” button to store the data in the system database.

We cannot delete information of an employee from the database because we associate this information with each pick-up and drop-off rental (see Figure 19 and Figure 20, respectively). If the system deletes an employee, the database will become inconsistent.

![Figure 22: First step for the managing employee data.](image)
Manage Cars and Location Information

When RenToGo.com acquires new cars or expand its business for servicing new locations, then administrator needs to update the system accordingly. When the administrator clicks on “Manage Cars and Location Information” button shown in the Administrator Main Menu screen, the system pops up the screen shown in Figure 24. Through this screen, the administrator can also insert new car types and add or update the daily rates of each car type. Observe that any of the information inserted and/or manipulated in this screen cannot be removed from the database for the same reason that we cannot delete employee data.
Generate Reports

Finally, when the administrator wants to check on how the company is doing, he can request the system to generate one of the following reports:

- For the last month, the average number of cars that were picked and returned to the same location, grouped by location and car type.

- For the last month, the average number of cars that were picked in one location and returned in another, grouped by car type. Also, indicate the location that the car was picked and the location that it was returned.

- For the last month, the average number of cars that were dropped off late, grouped by car type and location that received the car.

- Number of new registered users in each location in the last month.

- For each location, which car type was the most rented and which car type was the least rented in the previous month.

- Total sales for each car type, in each location in the previous month.

Similar reports will be required for other aspects of the system, such as hotel reservations, travel profile aggregation, etc.

PHASE II -- DATABASE DESIGN REQUIREMENTS

In Phase II, you will need to design the relational schemas for the database, and design the data sets that you use to populate the database for the testing and the demo. Note that to provide a common basis for all teams, I may provide you with a common Phase I solution. You are expected to populate this database by sets of data either downloaded from the Web or created by your team so that you can demonstrate the functionality developed in Phase III to your customer (me). The Phase II report must contain a description of the purpose of this project. It must attach with the Phase I report and must describe any revisions made to the specification described in the Phase I report. It must describe the problems encountered in this phase, and justify the solution. It must contain all of the documentation produced in this phase, including the development of (a) the set of assumptions for your ER diagram in terms of database content and dependencies, (b) a relational schema from the entity relationship model produced in Phase I, (c) the set of functional dependencies used to evaluate if the resulting schema is a BCNF or a 3NF; (d) the normalization process (if necessary) that transforms the relational schema to a relational schema of the BCNF or 3NF. In addition, if you have done any work in terms of actually loading the schema and/or tuples, please document as appropriate.

PHASE III -- APPLICATION DESIGN AND DEVELOPMENT REQUIREMENTS

GetAwayCheap.com is interested in a set of application programmer interfaces (APIs) that are easy to use so that they do not have to spend unnecessary amount of time to train their employees. To meet this requirement, you are expected to write at least a menu-based main program that consists of the following four application programs for their daily transactions and is open for the inclusion of future programs. The APIs that are for both customers and employees can have two distinct interface - web-based for customers and a Java application for employees. Overall, the interfaces that you design and implement for customers and employees can be minimal, to allow you to focus on the more critical database interactions; however, each of you should be sufficiently versed in programming (html or Java) to be able to rapidly prototype these GUIs. To assist you in

## Clients and Software Architecture

![Diagram showing clients and software architecture]

- **Web Client** (Customer)
- **Java Client** (Employee)
- **Query Program**

### Connect and Interact with any DBMS
- Personal Oracle
- MS Access
- MySQL

**Front half contains API (Java calls) that hide DB interactions from Client**

**Back half is the implementation of the classes that contains code to open the database connection, perform a query, collect results, etc.**

### FINAL REPORT REQUIREMENTS

The final report, due at end of the third phase, should be an integrated report. For the list of relational instances you use to populate your database, you should take data from a realistic situation, and make sure that your data shows some variety. The report for Phase III must include the following six components:

1. A description of the purpose of this phase of the project and the list of tasks, a summary of the system requirements and any additions or any revisions made to the specification and the design in the previous two phases,

2. The description of the list of problems encountered and your solutions.

3. The system architecture, and the design and description of all functions, including the list of required ones and the list of functions that your group add into the GetAwayCheap.com system.

4. The conclusion that evaluates the system you have implemented, the current limitations and the potential for improvements.

5. Each team member will provide an updated description on your contributions to the group project and what you learned from this project and the team work. This is the individual contributions on the web page. In addition, each of you must confidentially submit the final self assessment by email to steve@engr.uconn.edu without consulting any of your team members. This information is advisory only!
6. The collection of all of the submittals for the project, including: Phase I report, Phase II report, and Phase III report which includes (1) the relational schema in SQL DDL, (2) the source code of the four application programs, and (3) the relational instances you use to populate the database, (4) a sample output of the GetAwayCheap.com query and update programs, including the summarization outputs, and an attachment of a user manual for the system.

A hard copy of all the above components is expected at the demo time or on 5/01 in class. Together with the hard copy of your final project report, you need to turn in the report and the source code in electronic form using a winziped file. A more precise rundown of the final requirement will be given two weeks before the project is due.

**GRADING POLICY**

The following aspects are important for the grading: clarity of description and diagrams; completeness and sophistication of the ER model, the database design, and the data flow diagram; and coverage and quality of application program design and implementation, including correctness of all functions, friendliness of the API interfaces, and the clarity of user-manual. A generally acceptable report and/or implementation will get an "average score". A report with more complicated database design or application design will get more. An implementation with more complicated database or application or more user-friendly interface will get more. A report that only minimally implements the requirements in this specification for the three Phases will get a "below average" score. Please be aware that the database functions are more important and will take the larger portion of the grade.