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 UConnJobSearch.com, modeled after monster.com, that is being set up with the objective of providing the best services to students in their job hunt. 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, 2004, Finalize Groups for the Project
- Project Phase I: Monday October 4
- Project Phase II: Monday October 18
- Project Phase III: Monday, December 6 (or Wednesday, December 8) with Demos on Thursday and Friday, December 9 and 10.

PROJECT GUIDELINES

1. This project is a team project. The group size will be between 3 or 4 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 I: 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 diagrams) for a functional analysis of UConnJobSearch.com and conceptual modeling using the ER design model (required) and business process modeling (as needed). For your ER diagram, make sure you identify the critical assumptions and any relevant notes that characterize your solution. See the solution for the Spring 2003 project for an example Phase I solution for an ER diagram. All of your modeling/design tasks can be accomplished via the UML tool Together CC, which has all of these diagrams, and under the "Print" option, will generated a printed copy of all of your documentation. In addition, you are also required to create mockups for Job Posters (Companies, Universities, Organizations, Government Agencies, etc.) who are interested in position positions on UconnJobSearch.com.

   - **Phase II: Database Design** The main task of this phase is to map the ER model that you have designed for Phase I to the relational data model and to normalize the relational schemas into either third normal form or BCNF. In this phase, depending on the outcome of Phase I, you will either be asked to use your Phase I solution, or to use the common solution.

   - **Phase III: 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 GUIs, 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 UConnJobSearch.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 UConnJobSearch is rather generic and minimal 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 9 (1:00pm - 5:00pm) and December 10 (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. The first two reports are primarily to insure that your project design is on the right track. The graded design report for Phases I/II will be returned to you no later than a week from the due date.

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 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.

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 UConnJobSearch.com, a new job search website 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. 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 UConnJobSearch.com in achieving its goals. A preliminary requirements analysis has been conducted below that has identified a number of characteristics and features the operations (business processes) and goals of UConnJobSearch.com, including screen mockups that illustrate portions of the functionality. You, as the systems analyst/designer/engineer, are strongly encouraged to add, expand, and refine these requirements in order to achieve a richer design. Your deliverables in Phase I are: ER and UML diagrams (with assumptions), mockups of sample screens for the Job Posting Functionality (to post positions), and any written documentation that is necessary to describe your solution (document your ER and UML diagrams) and explain the functionality of your screen mockups.

For this project, you will develop an information management system that supports some of the services involved in a job search database (e.g., monster.com). In the following pages, the functionality of UConnJobSearch is illustrated using figures and a discussion of the flow that exists to take a user from screen to screen. Remember, these screen shots contain the base-line functionality for UConnJobSearch and can be enhanced with additional capabilities as part of your requirements analyses and your solution choices. When a job seeker accesses the application, the first screen that they see is depicted Figure 1. If the job seeker is an existing job seeker of the system they will enter their username and password then click “Submit” to access the system. If the job seeker enters an incorrect job username and password an error message pops up as shown in Figure 2.
New Job Seeker
If the job seeker clicks the new job seeker button, they will see the registration screen shown in Figure 3. The job seeker needs to provide the system the following information: the username and password they wish to use to access the system, their first and last name, and their address. After completing the form the job seeker will click “Register.” It is important to note that before inserting the user, the system needs to verify that the username and e-mail does not already exist in the system (database). If the job seeker selects a job username or an email that is already in the system, an error message pops up as shown in Figure 4 and Figure 5 (don’t forget to also have a combination error message for duplicate user names and emails). The job seeker should then be given the opportunity to enter another username or e-mail on the registration screen. If the job seeker decides not to register and clicks “Cancel”, return to the login screen (Figure 1).

Main Screen
The main reason for registering with UConnJobSearch.com is to allow the system to provide a more personalized environment for the job seeker whenever they use the system. Although there are a number of possible ways to personalize, for this project we will use only a simple one: every time a returning job seeker logs in, the system will provide a list of unfilled jobs that match their resume. In addition, the system will not display the jobs for which the job seeker has already submitted an application. The main screen is shown in Figure 6. At this point, the job seeker has 9 options.

1. Search: Perform a search (Figure 7), which will bring back list of jobs that match the search criteria and display them in Figure 6.
2. Manage Applications: View the list of applications that have been submitted by the job seeker. See Figure 8.
3. Update Resume: Allow the job seeker to update their resume. See Figure 9.
4. Update Profile: Allow the job seeker to update their profile. See Figure 15.
5. Apply to Selected: Add the selected jobs to the list of jobs for which the job seeker have applied. In a real situation, at this point the system would send the job seeker’s resume to the company that has posted the job. However, in this project we will only update the database to
reflect that the job seeker has applied for this job and remove the select jobs from the list in Figure 6.

6. **Select All**: Mark all check boxes.
7. **Clear All**: Clear all check boxes.
8. **Exit**: Exit the system, which will pop up the Initial Page (See Figure 1).
9. **View Job Detail**: The job seeker will be able to view the details of a job by clicking on the text of the Job Title. (See Figure 16)

---

**Search**

Let’s say that the job seeker wants to search the database for jobs that may not match their resume. They click “Search” and the screen in Figure 7 pops up. The job seeker then fills out the fields accordingly, and hits the “Submit” button. It is not necessary for the job seeker to enter data for all the fields. When the system finishes executing the search, the screen depicted in Figure 6 pops up with a list of unfilled jobs that match all the inputted search criteria. Once again, the system will not display the jobs for which the job seeker has already submitted an application. If the job seeker does not enter any data in any of the fields, and click on the “Submit” button, the system will bring back the entire list of unfilled jobs. As shown in Figure 7, a limited set of search criteria have been specified (one title, one company name, on salary, one location). For the project, at a minimum, you should expand this to include: multiple job titles (drop down or selection), salary range (minimum to maximum), and multiple locations (drop down or selection for all locations of that company). Do not search across multiple companies. These requirements impact on your database content – so carefully consider what you need. For example, the Job Titles represents a database entity that would contain all of the possible titles for all possible jobs – this is a rather static table (doesn’t change very often), but it is very useful for automatically filling a drop down selection box with values form the database.
rather than having this hard coded in the software itself. These types of tables will be useful all over, for example, for the states, for the cities within a state, etc.

![Figure 7: Search Screen](image)

**Manage Applications**

The option “Manage Applications” is in Figure 8. There are two new features on this page, “Delete Selected” and the “Filled” column. “Delete Selected” allows the job seeker to remove some or all of the applications they have submitted. “Filled” allows the displays the status of the application in the system. The job seeker is able to view the detail of a job by clicking on the text of the “Job Title”. If the job seeker clicks on “Main Screen”, Figure 6 will pop up. In addition to the information that has been listed in the table, you need to add a column for

![Figure 8: Application Screen](image)
“Company” who has posted the position (job). This column should be after Job Title. The motivation for this column is that the same job (by title) can be listed by multiple companies (e.g., Programmer for Aetna and Programmer for Pratt & Whitney). In addition, you may want to consider an “internal” JobId that can be used to distinguish all of the jobs. Adding a JobId would allow you to distinguish between multiple jobs with in the same Job Title category at a single company. These considerations add flexibility to your overall database design. See the discussion under the Job Poster’s section near the end of this document for additional information.

Update Resume

When a job seeker graduates, leaves a job, gets a promotion, or learns a new skill, the job seeker needs to update the database accordingly, which brings up the following sequence of screens. The narrative for all of these screens covers the mockups shown in Figures 9 through 14. To begin, in Figure 9, the job seeker is able to update their objective and desired salary. In addition, the seeker can attach a file (PDF, Word, PS, etc.) that contains a full-text version of their resume. The desired salary field is used in Figure 6 to determine if a job’s starting salary is greater than or equal to the job seeker’s desired salary. To navigate among the screens, there are next and previous buttons (next on only the first screen, previous on only the last screen). Once the user has updated the first screen (Figure 9), s/he can proceed to the next screen by clicking the “Next” button. In addition, the cancel button on any of the screens allows a user to return to the “Main Screen” (See Figure 6) without making any additional changes.

The screen in Figure 10 is used to update the job seeker’s education. The degrees are displayed in descending order by the graduation date. The job seeker is able to insert (Figure 11) and delete entries. The “Degree” field and the “Degree Area” field are both used to determine if a job seeker satisfies the educational requirements for the job. Once the information has been submitted, the user is returned to screen in Figure 10. The new entry should be displayed on the
screen. Overall, these two screens allow the user to view and then change or update information that is associated with his/her education (High School, College, Graduate School, Technical Training Certificates, etc.). Note that School, Degree (BS, BA, MS, MFA, etc.), and Degree Area (Math, CS, Biology, English, etc.) are all set via dropdowns. These three fields in Figure 11 are all representative of information that can be put in tables in the database (model as separate entities) so that the database can be modified when there are any changes (rather than the application code). So, in Phase III, these should be filled by a database query.

Figure 10: Resume Update - Education

Figure 11: Education Entry Screen
The next stage in updating the resume is updating the work experience (Figure 12). The screen in Figure 12 is used to update the job seeker's work experience. The experiences are displayed in descending order by the start date. Notice that End date of null represents current employment. The job seeker is able to insert (Figure 13) and delete entries. The “Job Title” field is used to determine if a job seeker satisfies the experience requirements for the job. Once the information has been submitted, the user is returned to screen in Figure 12. The new experience should be displayed on the screen.

Figure 12: Resume Update - Experience

Figure 13: Experience Entry Screen
The screen for updating the job seeker’s skills is shown in Figure 14. For the purposes of the project, we will have a single skill set which includes both technical and business oriented skills. For example, a secretary would be able to check Microsoft Word and Access. The selected skills are used to determine if a job seeker satisfies the skill requirements for the job. Once the information has been submitted by clicking “Finished”, the user is returned to screen in Figure 6. A new query should then be executed to reflect all the changes made to the resume. Again, these skills should be loaded from a database table as necessary. So in Phase III, you may have to get somewhat tricky in your software to allow the skills to be displayed dynamically based on the number that are in the database.

Update Profile
Lastly, if the job seeker needs to update some of their information (e.g., provide a different address), they can do it by clicking on the “Update Profile” button, which will bring the screen shown in Figure 15. The only information that the job seeker cannot update from their profile is the username. When opening this screen, the system must display all the information that is available on file for this job seeker. The update will take place only if the job seeker clicks the “Update” button. If the customer clicks on “Cancel”, no update is made to the profile. Once the job seeker is done with the update, the system will return to the Main Screen (Figure 6).

View Job Detail
If the job seeker desires to view the detail of a job, they simply click the text of the “Job Title” displayed in Figure 6 and Figure 8. This will bring the screen shown in Figure 16. A job can be associated with multiple “Degree Types”, “Degree Areas”, and “Skills”. The system should also inform the user if the job is currently filled and the number of applications currently submitted for that job.
Summary of possible screen sequences in the system
Here is a summary of possible screen sequences for each of the two types of job seekers. This definition of flow should be very useful when you are designing your State Chart and Activity Diagrams in UML:
• **New Job Seeker**: Start by clicking “New User” on the Initial Screen (Figure 1) → Registration Screen (Figure 3) → Main Screen (Figure 6) → Update Resume (Figures 9 through 14) → Main Screen (Figure 6) → Search Screen (Figure 7) → Application Screen (Figure 8) → Main Screen (Figure 6) → Login Screen (Figure 1)

• **Returning Job Seeker**: Start by entering “Username” and “Password” on the Initial Screen (Figure 1) → Main Screen (Figure 6) → Update Profile (Figure 15) → Main Screen (Figure 6) → Application Screen (Figure 8) → Main Screen (Figure 6) → Update Resume (Figures 9 through 14) → Main Screen (Figure 6) → Search Screen (Figure 7) → Main Screen (Figure 6) → Login Screen (Figure 1)

---

**Job Seeker’s Profile**

For each registered job seeker, the system will maintain a profile with the following information: name, address, email, and resume (both data entered form with all of the values and a file version). The reason both are stored is to allow database queries that match positions to resume to occur against the stored data, while the entire resume can be “shipped” electronically to Companies if there is a match. Whenever a registered Job Seeker logs on, the system will list a number of unfilled jobs that fully match or partially match the job Seeker’s resume. The resume contains the information about the job seeker’s education, experiences, and skills. See Figure 6. To keep the application simple, each job seeker will only have one resume in the system at any given time, but of course, may be applying for multiple positions.

---

**Job Poster’s Profile**

As part of Phase I, you are being asked to propose an interface for the **Job Posting System**, which will be used by companies to post jobs. This is a very critical interface, since it can be used to identify our database requirements in support of the posting, i.e., your database must be designed to handle its information. When putting together a mockup, you are to act as a business analyst and a database analyst in order to identify all of the pieces of information that need to be stored. Like for job seekers, there should be a logon and registration screen for companies, agencies, etc., who are posting jobs. There should be screens to allow the posters to define the parameters of the job (company, location, salary, duration, etc.). There should also be a screen to allow a text description of the job to be posted, as well as screens to take “payment”. Note that I am personally being somewhat vague for this part of the project – **Your responsibility is to propose a solution here that would be reasonable and consistent with the job seeker screens.** I am expecting both screen mockups and associated descriptions (similar to the job seeker mockups and description); these will influence your ER/UML designs.

The **Job Posting System** will be a completely separate application built on top of your database. For each registered job poster, the database will maintain a profile with the following information: username, password, name, address, email, company, and posted jobs. For security, a job poster can not be a job seeker. In addition, each job poster is associated with only one company. However, the job poster can post multiple jobs for that company. Job posters also notify the system when a job has been filled.

**Data**

The “Job Poster” and “Jobs” data files will be provided later. Each job will have the information displayed in Figure 16. Job seeker’s data and other data should be created by each team.
UConnJobSearch.com Personnel
In addition to job seekers and job posters, I am also asking you to consider the types of “reports” that UConnJobSearch.com personnel might be interested in executing against the stored database. For example, there might be report that summaries all of the available positions by job title (unfilled or filled). Alternatively, there might be report on all of the revenue that has been taken in by the job postings. Essentially, these are aggregation queries against the database. For Phase I, please provide a list of possible reports that may be of use for the personnel that are managing UConnJobSearch.com.

NOTE: Remainder of this file has not been completely edited – focus only on pages 1-12 for Phase I.

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. 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) a relational schema from the entity relationship model produced in Phase I, (b) the set of functional dependencies used to evaluate if the resulting schema is a BCNF or a 3NF; (c) 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
TO BE PROVIDED AT A LATER DATE.

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 UConnJobSearch.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 UConnJobSearch.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 winzipped 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 UML diagrams; 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.