

Due: Typed Team Roster by Wednesday August 30 prior to the start of class. If possible, teams should have a minimum of 9 and maximum of 10 members. Team rosters should contain the complete names of each member, preferred phone number, and email address. In addition, a list of at least one times per week the team has agreed they can meet for a minimum of one hour should appear on the roster. In forming teams, you need to consider compatible schedules, interest, and abilities. **All team members must do their share of developing the specification, designing, coding, testing, documenting, and formally presenting the project orally. No exceptions! You are encouraged to utilize whiteboards, Google Hangouts, BLOGS and other electronic media for some meetings including formats supporting voice/video.** Please select and utilize a team name. If you are not in a team by the deadline or have fewer than eight members anticipate me adding members to the team. Not all members of a team should expect the same grade. I have posted the semester project grades for students successfully completing the course last semester on my door and will leave them for at least another week.

A requirement of the course is to master evaluating the worth of contributions of all members of the team including you own. You will be forced to rank the worth of all member of the team. My evaluation of your progress in evaluating the performance of others will be part of your project grade. Your rankings of team members will in turn be used in determine part of their project grade.

The specification will be due prior to the start of class Wednesday September 20.

Submit the specification in hardcopy format and email a machine readable version to me. Remember a specification is a statement of what is desired, not how the desired will be accomplished.

Team specifications may be posted on Blackboard under Assignments. Your team may utilize its own specification for the design phase of the project or the specification submitted by another team. You may modify another team's specification if desired to complete the team project. The specification must be resubmitted with the design and again when the final project is submitted. Designs must also be resubmitted with the final product. Modification of the specification once design and implementation has commenced is discouraged (it will cost grade points).

Sample Specification Requirements:

The following is a sample group specification. As you are probably aware, sales have been static (flat) for the last six months. Your team has been selected to specify a new product with the goal of increasing our current market share of an existing market or opening a new market. After querying customers, Sales has found the customer base has some minimum technology requirements prior to investing additional capital in new software products. You are encouraged to use an IEEE or other organization formal presentation format.

Any new product must be transaction oriented utilizing a network effectively (local area or Internet). The design must utilize a three-tier database, i.e., a user interface, a separate shared processing element, and separate data storage element to implement persistence. We will be expected to utilize a database product. Basic operations should include whole record insertion, whole record deletion, and modification of appropriate values in existing records. Sales states the problem must be sufficiently complex to require multiple tables utilizing foreign keys. You must insure appropriate normalization to prevent insertion, deletion, and modification anomalies. The final implementation must be online with an N-Tier database, i.e., the solution must be multi-threaded (multi-processed) to receive an evaluation of "B" or better. All of our customers have moved to multi CPU hardware platforms and operating systems with concurrent customer access.

The specification should specify in detail all required inputs to the system, outputs, and special algorithms (e.g., formula or policy for overtime for hourly employees and salaried employees if appropriate). Sample input screens and reports should be provided as seen by the customer. The specification should contain sufficient information to support all major formal design strategies (see Preliminary Design Requirements for details). In addition to the required function, the specification should list any processing restrictions (e.g., support 100 transactions per second). The document should state the customer's acceptance criteria for the final product, e.g., MTBF of 5 months, MTTR of 2 days with a maximum of 4 days. It is assumed we will be responsible for lost data hence the product should contain automated data recovery facilities. If such facilities do not exist, there should at least be a comprehensive plan the customer can follow for backup and restoration of lost transactions. Please consider full backups, partial backups, audit trails, multiple copies of important files kept on separate storage media (mirror images), etceteras. The customers have indicated preventing the loss of data and transactions is a very high priority. The customer expects the use of automated email, secure communications (encryption), real time conversations in the form of whiteboards (etceteras) and other modern technology.

Including data flow diagrams (static and/or dynamic – no factoring), objects (but not inheritance hierarchies), timing diagrams (scenarios), etcetera would be appropriate. Remember this is a specification. If you find yourself thinking in terms of implementation, you are on the wrong track.

The design will be due prior to the start of class Wednesday November 1 (we may have to adjust this date latter in the semester). Submit the design in hardcopy format and email a machine readable version to me. Remember, a design is a statement of how the desired will be accomplished.

Design Requirements:

- 1) You must submit a functional design, static data structured design, dynamic data structured design, and object oriented design. You may submit additional design approaches if desired. Your design document must compare and

contrast the designs using technical metrics to establish the good, the bad, and the ugly for each design. Metrics should include (but not be limited to) cohesion, coupling (data and control), Miller's Law, Gracianas' Law, factoring, scope of effect, scope of control, utilization of black boxes, and fan-in (etcetera). Clearly state which design has been selected for the implementation phase and why it was selected.

Did you write a proper specification? If all information required by the design and implementation team is available, you should not have to make changes or additions to the specification. This is a learning process. If you made a mistake in the specification (especially an omission), correct it now!

There will be a temptation to assign specific design methodologies to other team members and avoid learning the detailed methodologies yourself. Regular tests and especially the final will require you to be conversant at the detail level in all design methodologies. Expect one or more design methodologies on each test. I recommend you schedule a team "walk through" for each design to insure it meets the specification. The walk through will provide an important opportunity to learn. All solutions should be equivalent.

- 2) Your design must be three-tier, i.e., a user interface, a separate processing element, and separate data storage element to implement persistence. You may utilize a database product or files.
- 3) Your implementation must be N-Tier, i.e., the solution must be multi-threaded (multi-processed) (for a grade of "B" or better).
- 4) You must explicitly indicate how race conditions have been avoided. Note this implies the specification is sufficiently challenging to create potential race conditions. If the specification is inadequate for the task, modify the specification. Include the problem description and solution in your design document (for a grade of "B" or better).
- 5) Explicitly explain all potential deadlocks for data access and how you avoid them to obtain a grade of "B" or better. Again, if your original specification is insufficient to exhibit deadlock modify the specification. Trivial specifications do not meet minimum requirements. For a deadlock to occur, you must have at least two related documents over which separate processes/threads may compete, e.g., an invoice and related purchase order, shipping record, or inventory transaction. Include the problem description and solution in your design document. Indicate how to avoid your specific deadlock using database or file system features (discuss another product if the database used your implementation does not have deadlock control), avoidance via programming, and management techniques (consider job scheduling).

- 6) Supply Entity-Relationship diagrams for your data showing all data, keys, foreign keys, one-to-one, one-to-many, and many-to-many relationships. As part of your design document, explain specifically why your database or file system is at least in third normal form. Boyce-Codd normal form or higher would be preferred (to attain a grade of “B” or better).
- 7) Network security is important. During your project demonstration you will be expected to create a private/public key pair. You must export the public key and import it into client programs. All network communications must be encrypted. You may use Secure-Socket-Layers, Transport Layer Security or a related technology if desired as opposed to encrypting and decrypting the communications yourself. Regardless of the technique, you must create and exchange appropriate keys to attain a grade of “B” or better. You may utilize SSL or TLS technology if desired.
- 8) As part of the design, consider user Remote Procedure Calls (RPC), Servlets, and multithreaded Sockets for implementation. Convenience me you understand the difference between technologies. Explain why you have selected the technology you will utilize during the implementation phase.
- 9) Your design document must specify how the final product will be tested and how the customer will measure the result for acceptance.
- 10) Your design document should indicate the potential for optimization for space and optimization for time.
- 11) Indicate specifically the parts of the specification and design that will lend their selves to re-use.
- 12) Your design must include an implementation plan to complete the project and present the results this semester.

Sample multi-threaded implementations using database technology on networks may be found at T:\CSC\DSB in the Java directory in several locations, e.g., BurrisJavaMultiThread.doc, BurrisJavaServlets.doc, BurrisJavaServlets2.doc, BurrisRMIServier.doc, BurrisJavaCookies.doc, and in the directories CS278JavaDB, and JavaRMI. You are encouraged to use a production language during implementation but lesser languages such a Perl (a real jewel) or PHP (if you are into the drug scene) may be utilized. Surely you are better than a snake charmer (Python). Use of Groovy with Grails or Ruby on Rails may make your project go faster and provide desirable experience for those going into business data processing. There are a significant number of jobs available in the Houston area using Ruby and Groovy.

It may not be possible to implement the entire specification during the implementation phase. We will discuss several means in class to mitigate this problem.

Implementation Requirement:

The implementation of your design will be due prior to the start of class Monday May 1 with appropriate support materials. Anticipate formal presentations of your working project November 27 with each team having 15 minutes. You must include your specification (possibly amended), designs, and code as a complete package. An exam will be administered the last week of November by ABET covering Software Engineering and all CS core classes. It will not be part of your course grade but will be very important to SHSU retaining its accreditation.

Management will consider alternate semester projects so long as they contain the specified/sufficient technical features! You must receive permission to do an alternate project! I highly recommend projects of interest to the group. Projects in which you are interested are fun. Projects just to meet the class requirements tend to be boring and hard to complete. As an example, in the past some very good interactive games have been produced. Feel free to “copyright” your work. You may wish to sell an application for smart phones or other money making ventures as a group. You may wish to automate data collection and presentation of Physics or Chemistry labs with grading available securely over the web. Projects automating some other university function or off campus organization would be appropriate. Recent projects have used Google Glasses, GPS on the Android platform and other recent technology.

As part of the project you will be required to rate/grade the performance of other members of your group and class. These ratings will in turn be rated as part of your grade by the instructor. At a minimum, ratings will occur after specification, design, and oral project presentation. *You will be forced to evaluate the performance of your team members.* You will also be expected to rate the performance of your group in comparison to the quality and productivity of other groups especially at the end of the semester. *Failure to accurately rate the performance of others will severely impact your semester grade!*

Experience teaching this course repeatedly demonstrates the more detailed the specification the less time the group spends in design, implementation, debugging and reduced friction

between group members! Vague specifications lead to lengthy projects and worse, personnel problems within the group.