

Comprehensive Project - Project Specifications

The comprehensive project will involve the development of a Windows/Web multi-user intranet application. It will be a team project, with you selecting your team. This project is a comprehensive review of the learning you have acquired over the past two years. As such, the project will be marked on the design, functionality and workability of the product and the process you undertook to develop the product. Product will be evaluated against the specifications and requirements not the amount of effort.

NOTE: in addition to this spec sheet you must reference the use cases and business rules documents provided in order to be successful.

Each team will be required to develop an Administrative Support System for a company of your choosing. You must develop a company name, company logo, and corporate presence that you can use to project the company's image. The Administrative Support System will consist of 2 distinct front ends – both a Windows and a Web interface – that will utilize a primarily common set of stored procedures and DLLs. Each front end will provide the necessary functionality specified below.

The system will require the following elements:

1. Purchase Order Requisition

This application will be available to employees in both a Windows and a Web environment to request various items required to do their jobs.

- This segment of the system allows employees the opportunity to request those items they require to support their work
- The purchase order request, once submitted, awaits processing by the employee's supervisor, who has the authority to approve or deny items on the PO request.
- A transaction must be implemented to ensure that no purchase orders can exist without at least one item.

2. Human Resource Tracking

A. The Windows based Human Resource System will provide HR employees a means to enter, modify and inquire on employee personal information, employment information, payroll information in total or by pay period, and sick days.

- B. A Web interface will be used by all employees currently working for the company, to inquire on their personal and employment information, as well as their future pension payout. Allow the user to select the inquiry they are interested in.
- The employee can update their personal address and telephone information.
 - The employee can inquire on their payroll information including their annual salary and bonuses, salary and deductions paid to date, details of any bi-weekly pay period.
 - The employee can inquire on their pension payout.
 - The employee can inquire on sick days taken.

3. Test Data

- You must have good test data that you create to test all system requirements and business rules.

Work Breakdown requirements:

1. The project must be divided between the two team members.

**Team member A must complete requirements for the Purchase Order requirements.
Team member B must complete requirements for the Human Resources requirements.**

2. **All other requirements ARE SHARED** responsibilities among the team, so you can tackle them as you see fit but both partners will be responsible for them.
3. Team members may assist each other and share work. However, each member is ultimately responsible for their given segment of the project.
4. The teams must produce one final project that **merges** the requirements of each member **for each iteration** along the way and a final merge at the end prior to deployment. The project **MUST** function as one entity – not two separate projects. The team will have to coordinate the production of classes, tables, etc that other team members may require.
5. The members should assign one member responsible as DBA, while the other member is responsible for file management and backups.

The General Requirements Include:

1. Develop the project iteratively and the work from both partners needs to be integrated properly in each iteration. You must create the appropriate iteration plans. You **MUST** do a walkthrough with the client at the end of each iteration.
2. The project must use C# as the primary development language. Implement at least one DLL that you coded in VB.NET.
3. Use n-tier development architecture using C#.Net, VB.Net, ASP.Net, and SQL Server. Include a Data Access Layer DLL, a SQL Layer DLL, a Business Object Layer DLL, and two Presentation layers (both Windows and Web). All Business Objects should be programmatically broken into its various functional pieces in separate classes. Use Object Factories and at least one validation class. All code must be contained within various projects within a single solution. Be prepared to show code snippets when asked.
4. Incorporate explicit transactions where required.
5. Validation/business rules should not take place in the presentation tier.
6. Implement interfaces in the object layer to insulate your development efforts from your partner.
7. Document all class members using the “summary” XML comment.
8. Implement all database access via SQL stored procedures.
9. Implement concurrency for updates that return new timestamp values where necessary for complete state.
10. Implement JQuery and AJAX (this can't be the AJAX Control Toolkit).
11. The project should be run from outside the VS interface.
12. Implement the Windows front end as an MDI app. Use splash forms, about forms, toolstrips, menustrips, and status strips.
13. Implement error handling on the front end and in stored procedures.
14. Include design elements that display the corporate presence of your company.
15. Implement a common Master page to ensure a consistent look and feel for the web application.

Deliverables submitted throughout the project:

1. The inception plan is not required. All other iteration plans must be printed and given to the user just prior to each iteration.
2. Use cases – any proposed alterations to the original provided use cases based on discussion with the user must be approved and submitted.
3. Iteration class diagrams **MUST** have feedback from the appropriate instructor, and this must be done prior to coding that iteration.
4. Working code for each iteration –the user **MUST** give feedback on the working iteration (done and tested) before that iteration is considered done. Any required changes to the iteration application must be implemented at the start of the next iteration. All iteration deliverables are posted to mark the end of that iteration.
5. The following deliverables must be posted, printed and submitted for each iteration:
 - Iteration Plans

- Updated use cases (if any)
- Class diagrams
- ERD

Marking Distribution:

OOAD component 30%

Technical Presentation of completed product; all code must be posted before the first presentation

- Individual contribution 50%
- Team contribution 20%