Comprehensive Project - Project Specifications

The comprehensive project will involve the development of a Windows/Web/Android Mobile multiuser 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, business rules documents and end user feedback 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 3 distinct front ends –a Windows, Web and Android mobile interface – that will utilize a primarily common set of stored procedures and DLLs. A REST API will be required for mobile applications and optional Angular Web front end. 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, department information and send email reminders for supervisors to complete employee performance reviews.

B. A Web interface will be used by all employees currently working for the company, to inquire on their personal information, employment information and performance reviews. Allow the user to select the inquiry they are interested in.

- The employee can update their personal address and telephone information.
- Supervisor employees will see pending performance reviews to complete and allow the creation of performance reviews for their employees

C. A mobile Android application will be used by all employees to browse and search employees by name and department. The employee information will display and a means to initiate and email or phone call to the contact directory from the mobile application.

3. Test Data

You must have good test data that you can 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. Each team must use a private GIT repository to handle source code and merging. The technical instructors will be administrators on the repositories. Details to follow.
5. 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. Ensure regular small commits to GitHub to avoid large merge conflicts is strongly encouraged.
6. The members should assign one member responsible as DBA (proper versioning of DDL, DML statements and test data seeding), while the other member is responsible for GitHub source control.

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 plan by breaking sub-tasks down in the Jira Project Management System. You MUST do one or more walkthroughs with the clients prior to the end of each iteration.
2. The project must use C# as the primary development language. Mobile app will be Android using Kotlin.
3. Use n-tier development architecture using C#.Net and SQL Server. Include a Data Access Layer DLL, a Repository Layer DLL, a Business Logic/Service Layer DLL, a Model Layer DLL, Web Services Layer and three Presentation layers (Windows, Web and Mobile). The Web front end can be built using ASP.Net MVC or Angular. All Business Objects should be programmatically broken into its various functional pieces in separate classes. All code must be contained within various projects within a single solution. No ORM's (ie: Entity Framework) can be used. 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. Document all class members using the "summary" XML comment.
7. Implement all database access via SQL stored procedures.
8. Implement concurrency for updates that return new timestamp values where necessary for complete state.
9. Implement the Windows front end as an MDI app. Use splash forms, about forms, toolstrips, menustrips, and status strips.
10. Implement error handling on the front end and in stored procedures.
11. Include design elements that display the corporate presence of your company.
12. Ensure that the Windows, Web and Mobile front ends have a consistent look and feel.

Deliverables submitted throughout the project:

1. The inception plan using Jira and use case sub-task breakdown is required. These must be completed at the beginning of each iteration and MUST have feedback from the appropriate instructor. Jira use cases, sub-tasks and tasks must be maintained as you work on each item. Full traceability of your work must be visible within Jira. This is done for you, you simple set your issues to in progress and done.

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. Any use cases not completed must be carried over to the next iteration in Jira. Iterations are determined done when technical instructors deem all use cases meet the **definition of done.** See below. Iterations are to be include, along with code, artifacts and process, a video demonstration of the iteration use case features must be recorded and submitted. Each submission will be evaluated by the instructor team and teams will be given the go, no go for the next iteration.
5. The following deliverables must be completed for each iteration:
   - Iteration plan using Jira Use Case sub-task breakdown
   - Updated use cases (if any)
   - Class diagrams (Artifacts checked into GitHub)
   - ERD (Artifacts checked into GitHub)
   - User meeting documentation (Attached to Jira use cases)
   - Code check in to GitHub for the iteration use cases.
   - Video functionality walk though demoing the use cases as per specified in Jira and clarification through end user interviews and feedback.

Definition of Done

The definition of done in terms of iterations is all use cases of the iteration contain all working features completed and thoroughly test. The use cases require all test data seed as well so any instructor can pull the repository and check any part of the system without having to generate test data.

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%