CMU 17-655 & 17-755 Architectures for Software Systems Spring 2014

Assignment 2: Call-Return and Tiered Architectures

Lattanze/Bass **Discuss:** Fri Mar 7, **Due:** Thu Mar 17

Problem Description

The objective of this assignment is to better understand call-return, client-server, and tiered architecture patterns that are so prevalent in the IT domain. Part of this assignment will be implementation-oriented. However, as with all of the assignments, this is not a programming class, so the emphasis of the assignment is on the architectural issues. Please keep in mind that this is not intended to be a complete IT application. Moderately sized IT applications in the real world can be millions of lines of code. Therefore the context for this system is intentionally small in order to a) keep the scope of the assignment manageable, b) limit the use of third party technologies, c) limit the complexity, and d) most importantly focus on architectural thinking. While IT technologies are not our focus, there are plenty of architectural issues relevant to the IT domain to wrestle with in this assignment.

The assignment consists of two parts. In part one your team will analyze the supplied architecture in the light of a given business context and decide how you will modify the architecture and applications that comprise the system. Part one will be done as a team.

The second part of the assignment consists of analyzing the architecture of the system in light of the modifications you made. After your analysis, design, and coding, you will reflect upon your work and answer questions related to the design decisions that your team made in part one. While you may discuss part two as a group, the write-up for part two must be done individually.

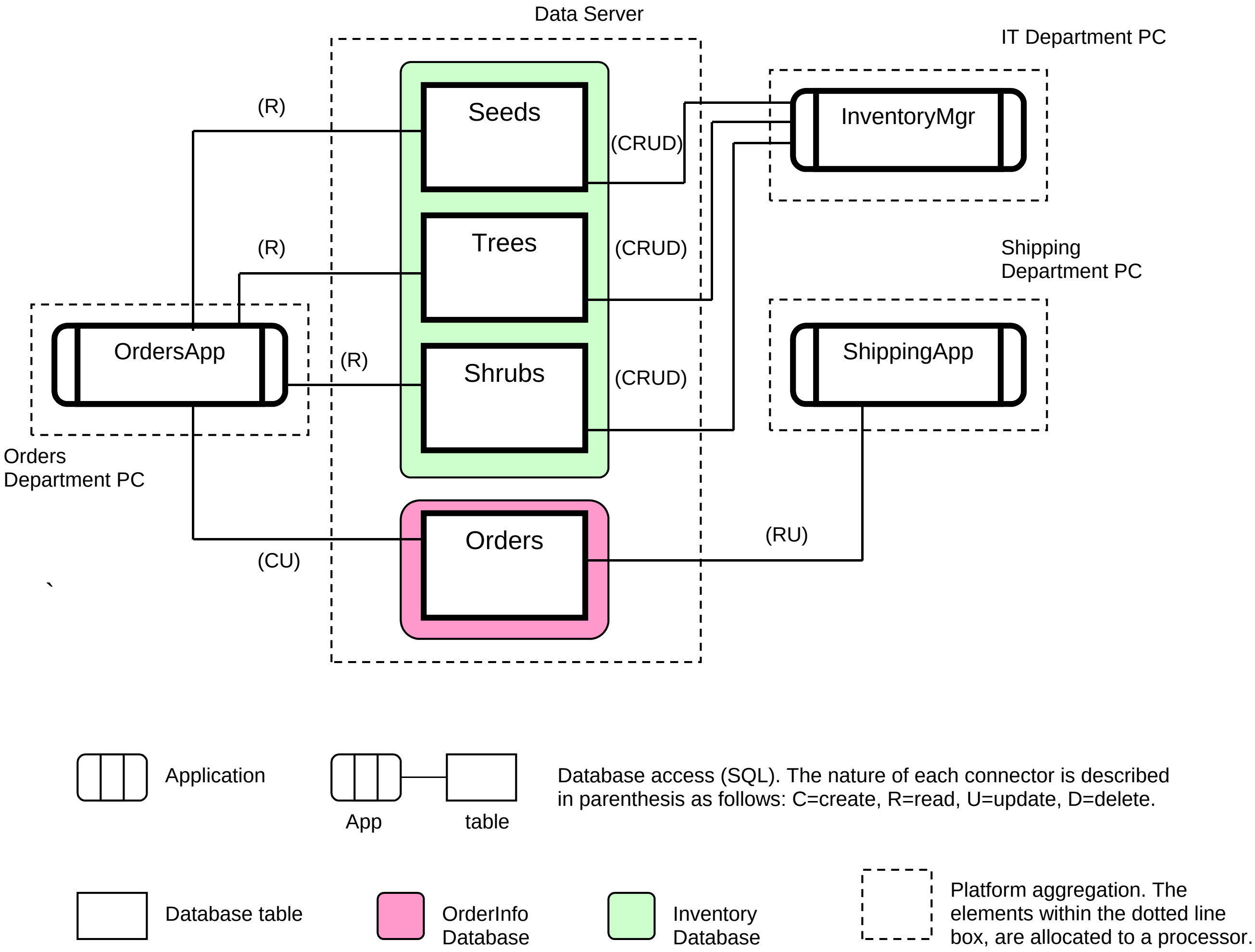
Business Context

You are the new IT manager for Exton Exotic Plants (EEP), which specializes in exotic trees, shrubs, seeds, and other plants used for the production and research of medications. EEP is a specialty business-to-business (B2B) company that supplies exotic and pharmacological plants primarily to medical and biological research labs around the world. EEP's business was founded through personal customer relationships but has grown considerably because of their reputation for very high quality products. Because of the specialty nature of their business, customers place their orders by directly calling EEP. Due to extraordinary growth, two years ago a simple database system was developed locally to streamline the work of the employees who take orders and enter them into the system. The system enabled the shipping department to read through pending orders, pack the order, and send it out to the customer. The EEP facility includes a large collection of green houses, labs, storage facilities for the products, an administration area where orders are taken, and an area where orders are prepared for shipping. All these are collocated on the same property.

The current system manages orders and shipping and was built by a consultant, who was under tremendous schedule pressure to get a basic order submission and tracking system on-line. The original requirements were to enable orders to be taken over the phone by EEPs personnel and enter them into the database. Once in the database, shipping personnel could then obtain a list of pending orders. Once an order was packaged and shipped, the shipping personnel could then "mark" the order as complete. The initial EEP database system provided two key applications:

OrdersApp for call center personnel, and *ShippingApp* for shipping personnel. In addition to orders management, the database maintains an inventory of trees, shrubs, and plants. The MySQL database tables were directly configured and managed by personnel using an inventory management program called *Inventory*. This is illustrated graphically by the diagram below:

Current EEP System Architecture – Dynamic Perspective.

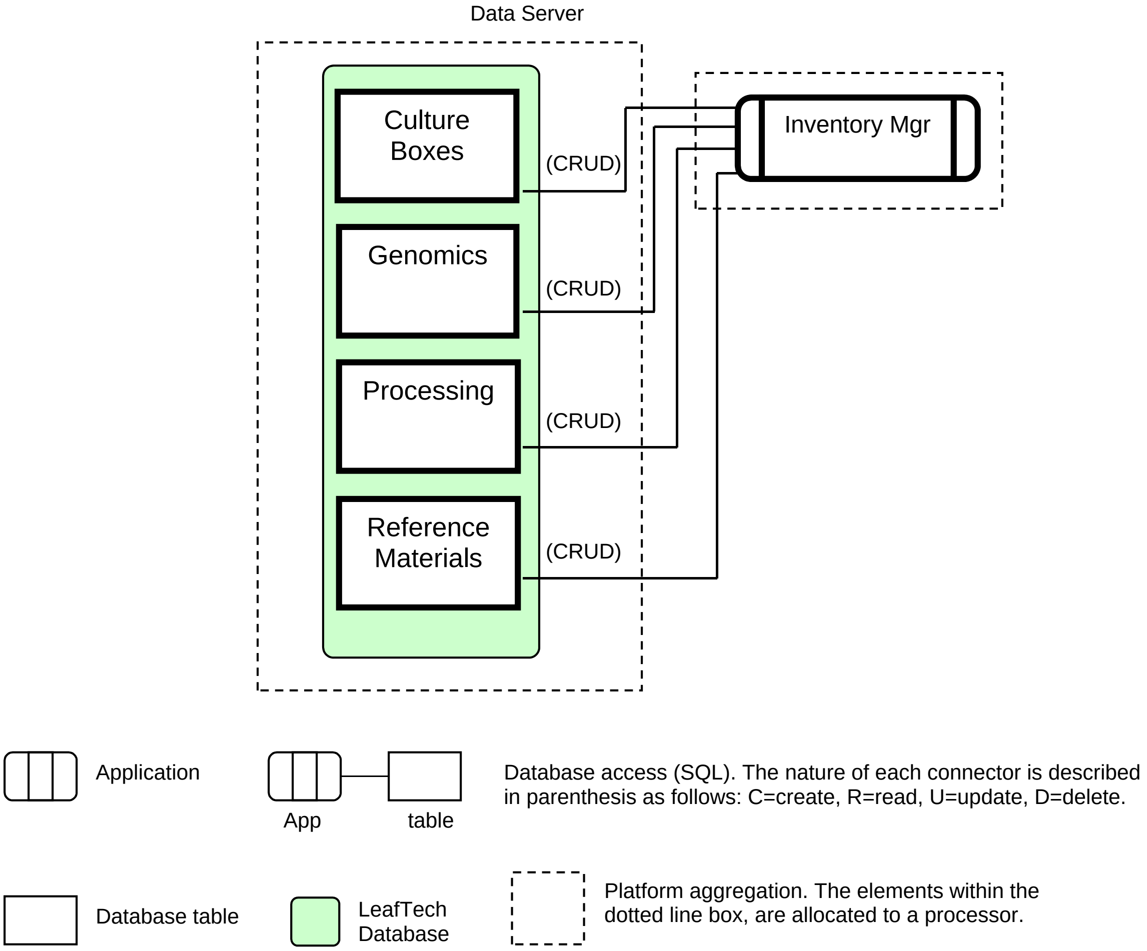


The current system was constructed around a client-server architecture, using a private local area network (LAN) where the client PCs are used in the administration building (for order entry and inventory management) and in the warehouse (for packing and shipping). The server is kept in the basement of the administration building. The system applications were built using the NetBeans framework where applications connect directly to the database via a local area network. The current system has worked reasonably well given the nature of their customers and their business models.

Recently EEP purchased a small lab supply company, Leaf Tech, which specializes in manufacturing specialty and custom lab supplies for pharmacology labs. They provide equipment to labs that specialize in plant-based pharmacological research. These labs frequently need to grow and care for exotic specialty plants and prepare them for study and processing. Leaf Tech was an attractive acquisition because like EEP, Leaf Tech is also a B2B manufacturing company with a specialized niche customer base with very unique needs. Leaf

Tech has developed highly personalized relationships with their customers and has a reputation of excellence in the narrow range of products they create. Leaf Tech also has a basic database that is used only to track inventory. As products are produced, they are entered into the database. Orders are taken by staff over the phone or from email. As products are sold and shipped, they are debited from the database using the same Inventory Management application. Their system is shown below:

Current Leaf Tech System Architecture – Dynamic Perspective.


Data Server
Culture Boxes
Genomics
Processing
Reference Materials
(CRUD)
(CRUD)
(CRUD)
(CRUD)
Inventory Mgr
Application
App
table
Database access (SQL). The nature of each connector is described in parenthesis as follows: C=create, R=read, U=update, D=delete.
Database table
LeafTech Database
Platform aggregation. The elements within the dotted line box, are allocated to a processor.

With the acquisition and merger of EEP-Leaf Tech the CEO would like to migrate to a real web-oriented system that would enable customers to browse products, place orders, track orders, and support all internal operations (e g. payment, shipping, inventory) and system management. This will be required to support the anticipated growth of company from a small regional company to a global organization. The near term goals are to support current operations and create a migration path to a system like the one described above.

Since the merger, managing inventory and tracking shipping for products has been a nightmare for the organization. The most pressing issues are 1) personnel must use multiple interfaces for Leaf Tech and EEP products, 2) the Leaf Tech system does not track orders, and 3) both systems are totally unsecure. The CEO of EEP-Leaf Tech has made fixing and updating the current data system a top-priority for you and your team. However, in the process of “fixing and updating" the current system, you must continue to support all current operations as they are and enable the organization to easily transition to a web oriented system. To be clear, priority one is that whatever modifications you make to fix the system in the short term must not interfere with user's ability to do their work today. Priority two is to ease the transition to a full web oriented system, again, in such a way as to not interfere with current operations. The CEO of EEPs-Leaf Tech realizes that a complete makeover of the system is impossible given these two priorities and has approved a transition plan that includes a number of phased activities as described below.

Part I: Modifications to the Current System (TEAM TASK)

Management's short-term priorities for the system are as follows:

1. Merge the Leaf Tech and EEP inventory management systems under one interface with a consistent look and feel.
2. Modify the system so that the shipping application also includes Leaf Tech products.
3. Provide authentication for all users.
4. Keep track of all log-in/out activity (username, log-in/out time). You do not need to utilize existing interfaces.

As you incorporate these changes into the system, assume that users will be using the original system until such time you migrate them to the new applications. From the user's standpoint, migration is an event… one instant in time they are using one of the legacy applications, the next instant they are using your improved application. This includes ensuring data consistency from the legacy application to the new application.

Installing the Source Code

Unzip the *A2 Source*.zip file into a working directory. You will find a document named *A2 Installation.pdf*. This document describes how to install the application, databases, JDBC Connectors, and the development framework.

Packaging and Submitting the Team Assignment (Part 1)

- You should submit a single system that incorporates all of the changes described above.
- Part 1 must be emailed in a zip file following the following format TEAM-NAME_ASSIGNMENT-NUMBER_YEAR to the course blackboard.
- We will test your systems on our computers. You may assume that we have Netbeans (7.4), MySQL(5.6.15), J connector(5.1.28), and databases populated with test data as described in the assignment (inventory, orders, leaftech). PLEASE DO NOT ASSUME THAT WE HAVE ANY OTHER RUNTIME ENVIRONMENTS INSTALLED ON OUR SYSTEMS. WE WILL NOT INSTALL OTHER RUNTIME ENVIRONMENTS OR SOFTWARE ON OUR SYSTEMS TO RUN YOUR PROGRAMS. You must clearly describe how to setup and run your program. It is fine to include other databases, software, and so forth, but you must clearly explain how to install, configure, set up and run your system. In general, we will not recompile your code on our machines. If we cannot figure out how to run your program, you will be penalized. If you have questions or concerns, please ask.

Part 2: Architectural Analysis (INDIVIDUAL TASK)

Even though the system design and modifications were made by the team, you are free to agree or disagree with your team's decisions in your individual write-up. If a question requires an architectural diagram, you are allowed to share the team diagrams, but you may modify and/or extend the diagram(s) as necessary to support your description, discussion, and analysis. Do not share any diagrams you create individually. Please answer the following questions **individually**. Each question has several parts - make sure you answer each part.

1. According to the business context, what are the key architectural drivers of the system and what are their relative priorities? Please describe those factors that influenced your design decisions.
2. For your modified system:
    - Provide views of the architecture of the modified system. Be sure to include appropriate views of the system. You are free to use whatever notations you prefer, but follow best documentation practices (per the guidelines discussed in class).
    - How has the overall structure (pattern) of the system changed and why? Discuss how well the design of your upgraded system supports the business goals of the organization in terms of the important architectural drivers you listed in Question 1.
    - What key tradeoffs did you make in the design of the upgraded system? Did you consider alternative approaches? Why did you select your approach over others?
    - Explain how your system was built without disrupting current operations.
3. Assume that you are making a case for the CEO of EEPs-Leaf Tech why your architecture for the future system is the best way forward. Define a road map for transitioning the current system to the full web oriented system that would allow customers to browse products, place orders, track orders, and support all internal operations such as taking payment, shipping, inventory, and general systems management. Your road map should answer the following questions:
    a. Give a detailed description (views) of the future system (structurally) and how it differs from the current system (the system you modified per the requirements in part 1). Describe how it works and how will meet the needs of the EEPs-Leaf Tech organization. Describe any architectural drivers that might change or emerge as more important and/or difficult to achieve as a result of transitioning to the future system.

    b. Create a transition plan that defines the necessary steps, priorities, and the logistics required to successfully develop and deploy the future system (again, without disrupting current operations). Provide a detailed schedule that articulates which legacy software applications, hardware and other infrastructure will be phased out. Explain how and when new hardware and infrastructure will be procured and installed; how and when applications will be developed and phased in. Include any other activities that you think are relevant to your transition plan. Be concise but thorough - the more thoughtful, direct, and realistic your plan is, the better. Your time line can be relative (e g. not sync'd with particular calendar dates).

Submitting the Individual Assignment

Post your individual assignment to BlackBoard in Word file format with the following naming convention: A2_INDV2014_<Andrew ID>. So my A2 assignment would be posted as A2_INDV2014_al45. **Please include your team name in your individual write-up to facilitate its cross-referencing with the team tasks.**

Grading Criteria

Your individual and team solutions and commentary will be graded upon:

- The quality and contents of your analysis in the individual write-up. Be clear, concise, and complete.
- The proper use of English prose, absence of grammar problems and misspellings.
- The consistency between the design representations and the implementation.
- Professionalism, which includes the quality of the team presentation, timely submission of team and individual assignments, and well-structured and documented source code.
- The correct operation of the solutions.

A2 will be weighted as follows (100 points maximum):

Part 1: Team Implementation: 50 points

Part 2: Individual Write-up: 50 points

- Question 1: 10 points
- Question 2: 15 points
- Question 3: 25 points