Proposed Design of an Inventory Database System at **Process Research ORTECH**

System Design

Prepared by

Andrew Ramadeen
Manojav Sridhar
Kunendran Deivendran
Junaid Yousuf

Monday, April 16, 2001

Introduction

The Company we are studying is Process Research ORTECH (PRO), a newly privatized company that was part of a large government research organization called ORTECH. PRO is part of the Metallurgy and Materials science research industry. They are a small organization with about 20 employees. Most of the work done at PRO is of a contract nature. When a client company approaches them and presents them with a problem, they analyze the problem and try to solve it on a small scale in their laboratories. If the problem is solved successfully and in a cost-effective manner, this solution is sold to the client for a negotiated price. Even though the organization is not a large one, the information that is required to carry out the research is enormous. PRO has a simple management structure. There is a board of directors who advise a company president. Under the president are scientific and office managers who oversee day-to-day operations of the company. And under them are the various scientists and office staff who do the experiments and run the front office. See Appendix 3 for an organizational chart.

Identifying the Problem

There are many steps involved in the process of solving a client's problem. The steps involve billing the client, purchasing materials, checking inventory and so on. Due to the rapid growth of the company, many standard procedures used to perform these tasks are becoming insufficient to meet the needs of the company and their clients.

We have already carried out a Feasibility Study and a Requirements Analysis at PRO and determined that one of the areas that needs the greatest attention is the inventory system. The management agrees that this area of their business requires immediate attention and they are actually considering the conclusions of our Feasibility Study.

The system was originally designed for a much smaller workforce, but with recent growth and workforce expansion, the system has become inadequate, thus impeding efficiency. This has resulted in relatively large project delays, inventory wastage and increased cost of maintaining the legacy system. For a complete description on the current system, please refer to Appendix 4.

Requirements

The first stage in designing a solution to this problem is to determine the requirements of the new system. A complete analysis was performed and documented in the Requirements Analysis document released March 12, 2001. What follows below is the list of functional and non-functional requirements for the new system generated by that study. For a description of each item, please refer to Appendices 5 and 6.

Functional Requirements

I. **Details of data storage**
   a. Inventory Items
      i. Name
      ii. Location
      iii. Usage
         1. Dates of usage
         2. Projects usage
         3. Personnel usage
         4. Amounts of usage
      iv. Date of Order
      v. Expiry date
      vi. Cost of item
   b. Scientists names
      i. Projects they are working on
      ii. Current orders they have placed
   c. Projects names
      i. Materials needed
      ii. Project start date
      iii. Project finish date
      iv. Supervisor

II. **Details of output**
   a. Inventory list screen
   b. Add new item screen
   c. Remove item screen
   d. Check availability screen
   e. Check location screen
   f. Check-out item screen
   g. Update item screen
   h. Place order screen
   i. Print order screen
   j. Print reports screen/Print hard copy
      i. Inventory Reports (various reports)
      ii. Personnel Reports (various reports)
      iii. Project Reports   (various reports)

III. Details of input
- a. Paper documents
    - i. Current inventory system info (paper files)
    - ii. Shipping order
        1. Item name
        2. Item amount
        3. Item date
        4. Persons name who ordered the item

IV. Details of Information Processing
- a. Inventory item
    - i. Usage details
        1. Amount remaining
        2. Time to expiry
        3. Amount used by project
        4. Amount used by Personnel
        5. Amount used MTD (month to date)
        6. Amount used YTD (year to date)
    - ii. 2. Cost details
        1. Dollar amount used by project
        2. Dollar amount used by personnel
        3. Dollar amount of item used MTD
        4. Dollar amount of item used YTD
- b. Personnel usage
    - i. Dollar amount of items used YTD
    - ii. Dollar amount of items used MTD
    - iii. Dollar amount of items used per project
- c. Project usage
    - i. Dollar amount of items used YTD
    - ii. Dollar amount of items used MTD

Nonfunctional Requirements

I. Software Requirements
- a. MS Windows 2000 Server
- b. MS Access
- c. MS C++

II. Hardware Requirements
- a. Pentium-II 500 MHz
- b. 300 MB hard disk space
- c. 256 MB RAM
- d. 10Base-T Network Interface Card
- e. Laser Printer

III. Security Requirements
- a. User name and Password identification for all users
- b. MS Windows Primary Domain Controller

IV. **Reliability/Survivability Requirements**
   a. Monday to Friday availability
   b. 8:00am – 6:00pm availability on those days
   c. Data restoration within 24 hours of data loss
   d. Daily/Nightly back-ups of database
V. **Interface Requirements**
   a. Simple interface
   b. No large user manual required
   c. Short training session
VI. **Lifecycle Requirements**
   a. System upgradeable
   b. Development time < 6 months
VII. **Economic Requirements**
   a. Approximately $50, 000 development cost
      i. Salaries
      ii. Software
      iii. Hardware
      iv. Installation

As we move through the design phase, each one of these requirements will be mentioned and dealt with by number.

Design Phase

There are 5 parts to the design phase as we present it here: a global system architecture, a diagrammatic modeling of the new system, a software architecture, a database design and a user interface design. After carefully documenting the current status of PRO in terms of its computer capabilities (see Appendix 9), we set out to complete the design phase.

1. Global System Architecture

In this section, we will be dealing solely with the hardware, networking and software platform infrastructure to be used in the new system.

Hardware

The current hardware situation at PRO is actually quite good (see Appendix 9). The machines are relatively new and more than adequate to handle the simple database solution we are proposing. The machines are all microcomputers, very open and very glass box. Hardware requirements for the new system can be found in the non-functional requirements, part II (see Appendix 6). It is clear to see that the current hardware at PRO matches or exceeds these specifications. The current server runs at 400 MHz, and has more than enough RAM and free hard

disk space for the new system. Each client machine has adequate processor power, RAM and hard disk space to use the new system as well. Thus simply maintaining PRO's computer infrastructure as it currently stands fulfills non-functional requirements IIa1, 2 and 3 as well as requirements IIb1 and 2. With 15 client machines, there are more than enough terminals for everyone to access one. They are present in all the offices, labs and work areas, thus there is no need to purchase more. With nightly 8 Gig tape backups, the database can be stored every night and retrieved with little or no hassle in case of data loss (non-functional requirements IVc and d). For a complete analysis of hardware criteria and alternatives, see Appendix 10.

Networking

The current network at PRO is set up as a simple peer-to-peer, TCP/IP Windows™ network. Currently, it is mainly used for email, the Internet, printing and file sharing (see Appendix 9). The cables are copper Ethernet, capable of transmitting at 100Mbps. The hub supports 32 users and 100Mbps, thus there is more than enough room and bandwidth for all 15 machines to be accessing the system at once. The machines are an acceptable distance apart, and the workload can be split and efficiently channeled by the switched hub. The new system can use the exiting network to connect to the database (using TCP/IP), query the database and return results. In addition, there is a network printer (an HP Laser Jet) accessible from anywhere in the LAN. Thus non-functional requirements IId and e and VIIa3 are satisfied by the current setup with no need to upgrade (see Appendix 6). The current network has an acceptable amount of downtime and since it has been in use for 2 years, we know it is reliable (non-functional requirements IVa and b). For a complete analysis of networking criteria and alternatives, see Appendix 11.

Software Platform

The current software situation at PRO is very impressive (see Appendix 9). They have Windows 200 Server running on their server and Windows 2000 professional running on every client machine. They have MS Office Professional on every machine, which includes Access (the database program which we will use to construct our inventory database). This fulfills non-functional requirements Ia and b (see Appendix 6). The only additional piece of software required is MS C++ which will be used to design the user interface. A license and a copy of the software can be obtained form any one of a number of authorized Microsoft distributors for about $100 dollars, which is still well within the economic limits set out by section VII in the non-functional requirements (see VIIa3). Microsoft software is very widely used, and there is a plethora of documentation and

cheap technical support for it. Also, it is relatively safe to assume that Microsoft will be in business for some time to come. Thus we can be sure of continued technical support, and frequent software upgrades (non-functional requirement VIa). For a complete analysis of software platform criteria and alternatives, see Appendix 12.

With the remarkable current situation at PRO, very little is still required for the new system. Most of the requirements are met or exceeded by the current hardware or software, and there is only one piece of software that needs to be purchased. Thus the global system architecture will incorporate almost exclusively items already present at PRO: (see Appendix 13)

Hardware

    a. Server – Microcomputer, open, glass box.
    b. Pentium II @ 400 MHz.
    c. 256 MB DRAM
    d. 8 Gig HDD (currently 6 Gig free space)
    e. 8 Gig Tape backup
    f.   10/100-BaseT NICs
    g. 32X CD Reader
    h. 4X CD Writer
    i.   1.44 FDD

Clients – Microcomputers, open, glass box

    a. Pentium @ (100-200) MHz.
    b. 32-64 MB RAM
    c. 2-4 Gig HDD
    d. 12-32X CD Reader
    e. 10/100-BaseT Network Interface Cards (NIC)

Network – LAN

    a. 100 Base-T twisted pair Cat. 5 cable (copper)
    b. 32 port, switched hub
    c. HP LaserJet 5P Printer & Jet-Direct LAN interface

Software – MS Windows and MS Applications

    a. Microsoft Windows 2000 Server
    b. Microsoft Office 2000 Professional
    c. Microsoft Windows 2000 Professional
    d. NetBEUI Protocol
    e. TCP/IP Protocol
    f.   Windows Primary Domain Controller

g. *Microsoft C++ Foundation Classes (must be purchased)*

2. Diagrammatic Modeling

Now that we have all hardware issues settled and a software platform upon which to build this system, we need to turn our attention to the software design. The following is a series of diagrams that show the system in various modes of operation. Some model the system as a whole, some model specific parts. Together, they provide an accurate picture of the system we intend to build.

Class Diagram:
These diagrams are useful in viewing the system as a whole. This diagram has been adapted from the Requirements Analysis and shows all actors and internal objects used in the system. Lines are used to indicate relationships between items. For a complete description of this diagram see Appendix 15.
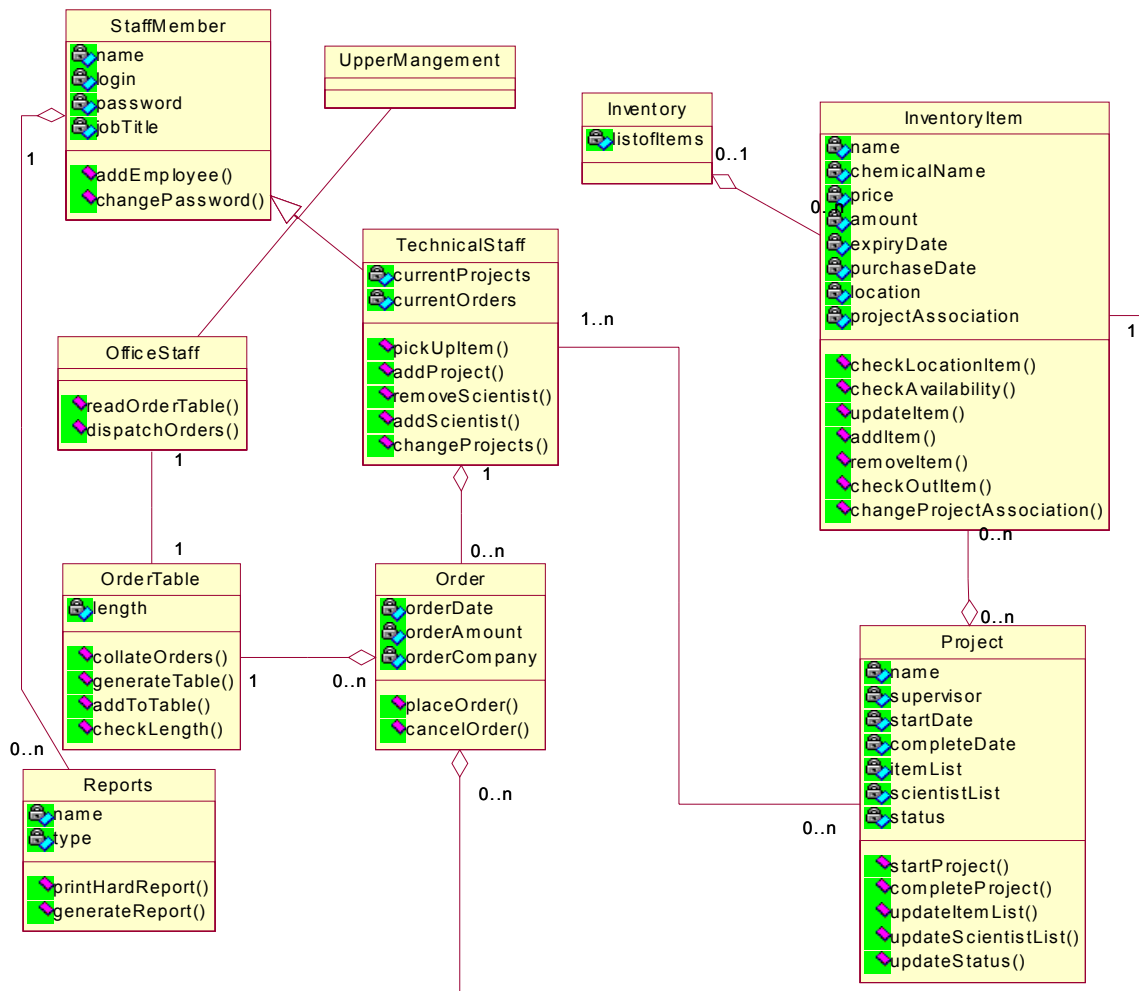
State Diagrams

State Diagrams have been used here to model the behaviour of internal system objects. These are objects created by the system such as tables, lists, etc. They have definite 'states' and can move from state to state after the occurrence of specific events. The black circle and the black bull's-eye represent the start and stop states respectively.

Class Inventory Item – see Appendix 16
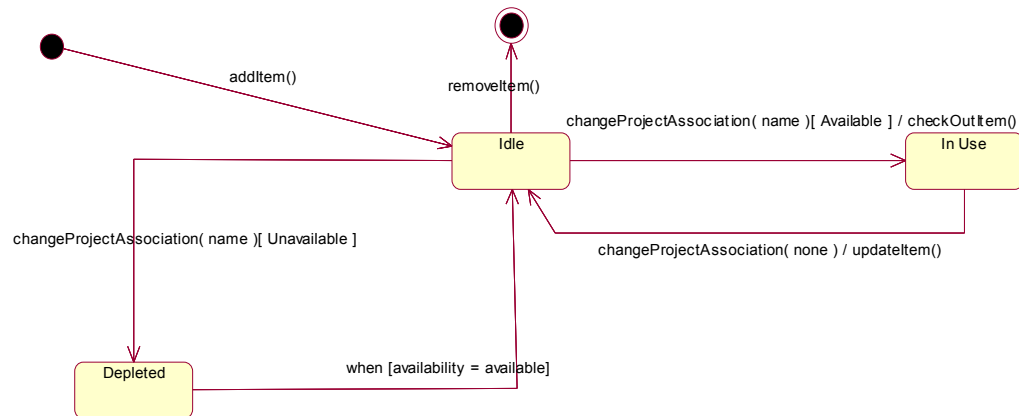
Class Order Table – see Appendix 18
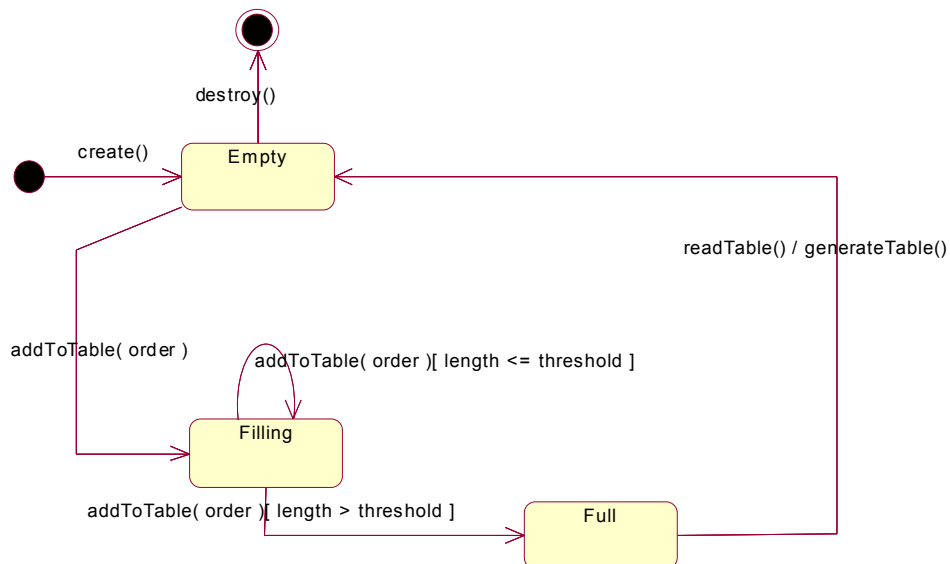
Class Project – see Appendix 19
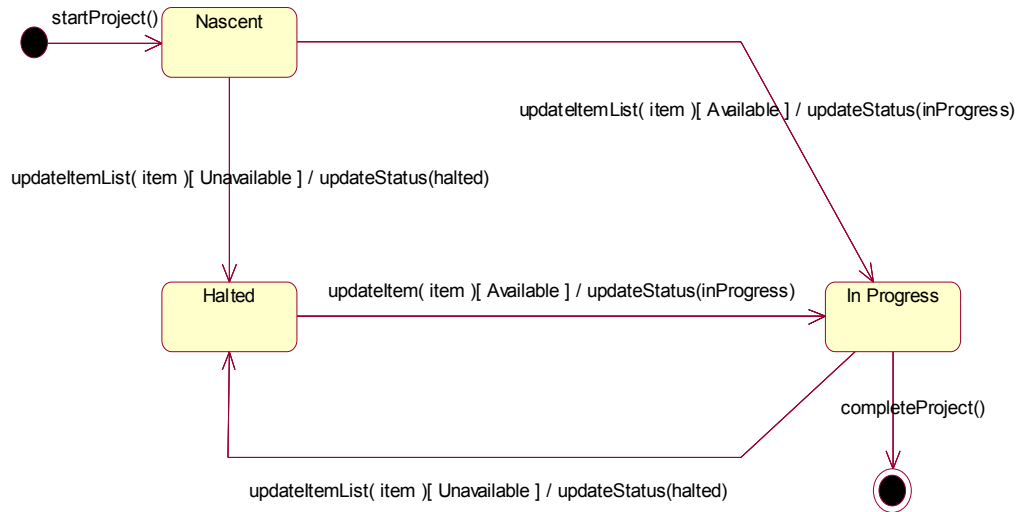
Activity Diagrams

These diagrams are similar to state diagrams except that they are more useful in modeling the actions of humans interacting with the system. Activity diagrams show the step-by-step movement of actors as they react to changes in the system brought about by previous actions, or by the actions of other objects or actors in the system. The black circle and the black bull's-eye represent the start and stop states respectively.

Class Office Staff – see Appendix 17
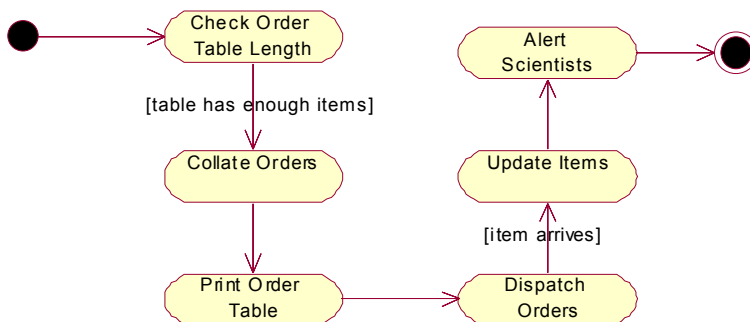
Class Technical Staff – see Appendix 20
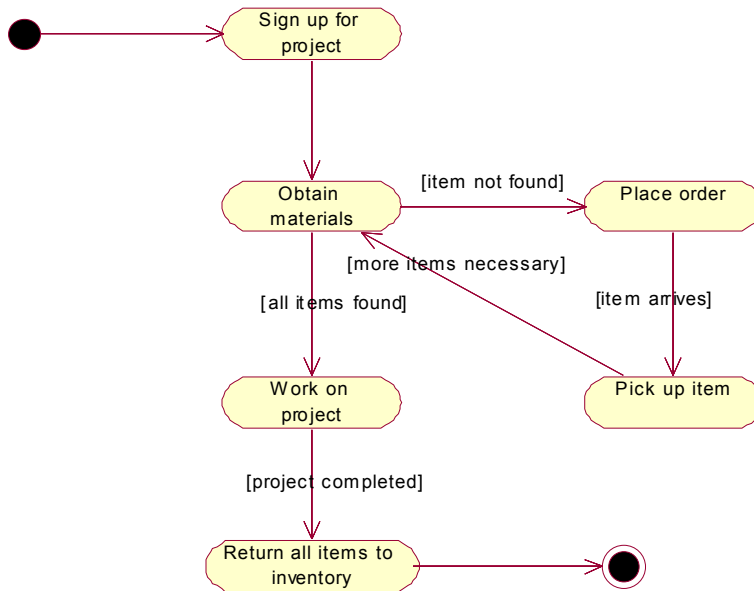
E-R Diagram

The E-R Diagram is similar to the class diagram in that it models the system as a whole. It shows entities, which are similar to the classes in the class diagram. It shows the attributes of each entity and the relations between entities (also present in the class diagram). This diagram is modeled directly from the class diagram seen above. Once we begin database design, this diagram will change to a form that can be represented in a relational database. For details on this diagram including a data dictionary and business rules, refer to Appendix 21.
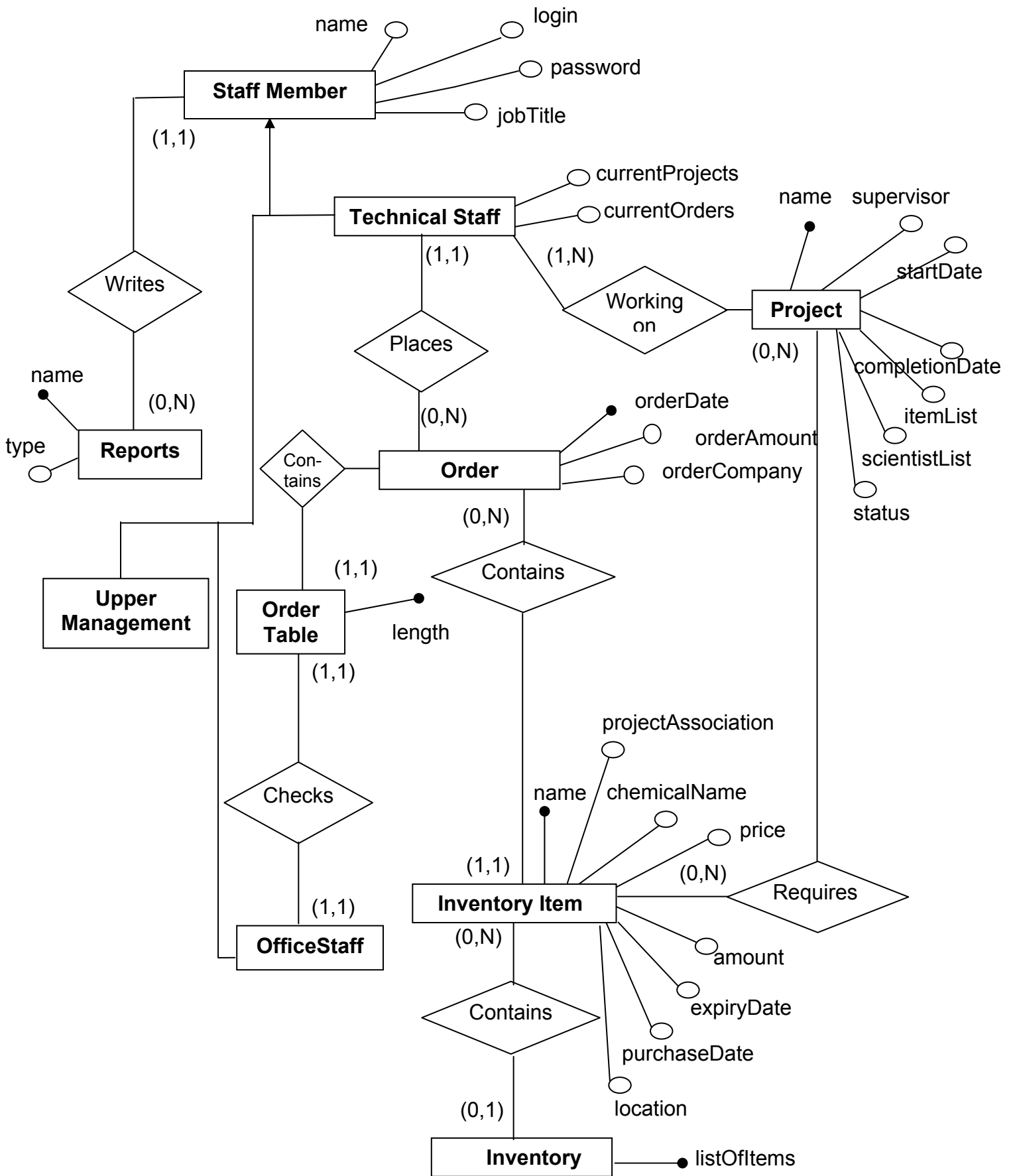
3. Software Architecture

With a pictorial view of the software we wanted to build, we set about to actually design it. Although we considered many possible architectures, there were two alternatives that our team debated: the Three-tier Client Server Architecture and the Repository-Based Architecture. They both seemed fairly appropriate, so we present them both here:

Three-Tier Design

The three-tier model is a type of client server architecture. This model consists of two types of machines, service consumers (clients) and service providers (servers). What is most important about this architecture is that clients and servers may or may not be running on dedicated machines. Thus it is possible for a machine to act as both a client and a server depending on the situation. In the case of PRO, the clients will be the individual microcomputers all around the company and the central server will be the service provider. Information exchange between the machines is done through messages requesting data, or invoking procedures on the remote machine. In this case it will usually involve one of the clients requesting information from the server. The three tiers of the system will separate the user (client) from the complex inner workings of the software. The first tier is an application layer that the user interacts with. Under the application layer is a processing layer that processes the user's queries. Under that is the database layer which receives the processed queries and returns the appropriate data to the processing layer. The processing layer passes the data back to the application layer which formats it for the user to see. The layers will exist on different machines; the processing and database layers will exist on the server, while the application layer will exist on each client machine (see Appendix 22).

Repository-Based Design

Repository based architecture is very similar to the client-server model described above. The system revolves around a central data structure (usually a database), and a collection of independent components which operate on the central data structure. One of the only main differences between this and the three-tier model is that in this case there is a dedicated server and dedicated clients. Under no circumstances can a client machine become a service provider, thus the system is very directed and straightforward. In the case of PRO, the central data structure would be the inventory database, which would be located on the server. Each client would access the server over the network and query the data they require. All software procedures would be present on the server and accessible from any client machine (see Appendix 22).

We decided that a repository-based system would be most appropriate for the software we are trying to design. The central data structure would be the inventory database and there would be multiple dedicated client machines all capable of accessing this repository. The database could be designed and installed directly onto the server and be instantly accessible from any computer on the network. We felt that there was no need to complicate the situation by creating multiple tiers and having any computer capable of being the service provider. This "one-way" access system also provides a level of security (since no one will have access to any other machine on the network except the server). It also greatly simplifies the construction of the system for the developers and the use of the system for PRO employees.

4. Database Design

Since the Feasibility Study phase, it has been our intention to design a relational database with MS Access. Thus our database will be composed of a series of tables containing the data we require. We will indeed use MS Access to design these tables and relations (for software platform decisions, see Appendix 12). We did look at the hierarchical and network models, but quickly rejected them both. Our data did not fit nicely into the hierarchical format i.e. there was no clear 'parent-child' link between any of our data. The network model was a possibility, but the implementation seemed too complex and the relational model provided a much simpler alternative.

Once we had selected the appropriate architecture, we analyzed the E-R Diagram shown above. For a complete description of that analysis refer to Appendix 25. After our analysis, there were many changes to consider. First we had a few redundancies to deal with in our initial design. For example, projects kept a list of inventory items they used, and inventory items kept track of their project associations. Some redundancies have been removed now, but a few still remain. In many cases, we felt that these redundancies were best left in place i.e. their presence will make the system work faster (quicker data search and retrieval times). The storage space they consume is negligible considering that we have reserved 300MB of disk space for this database (much more than a database of this magnitude requires). The generalization relationship between the staff members was also eliminated. Since different types of staff members had different operations, it was impossible to combine then into a single entity. Therefore they were left as separate entities and each one is pictured with its own set of staff member attributes. No partitions or mergers were performed on any of the entities. It was felt that each entity represented much too distinct a concept for there to be mergers. On the other hand, each entity was very concisely described, so no partitions were thought necessary. With this analysis complete, the new ER diagram (shown below) was created.

Relational Model:

Technical Staff (<u>login</u>, <u>password</u>, jobTitle, currentProjects, currentOrders)

Order(<u>orderDate</u>, <u>technicalStaff</u>, <u>inventoryItem</u>, orderAmount, orderCompany)

Reports(<u>name</u>, type, author)

Project(<u>name</u>, supervisor, startDate, completionDate, status, inventoryItem, technicalStaff)

Order Table(<u>order</u>, length)

Inventory Item(name, chemicalName, price, amount, expiryDate, purchaseDate, location)

Office Staff(<u>login</u>, <u>password</u>, name, jobTitle)

Inventory(<u>inventoryItem</u>)

UpperManagement(login, password, name, jobTitle)
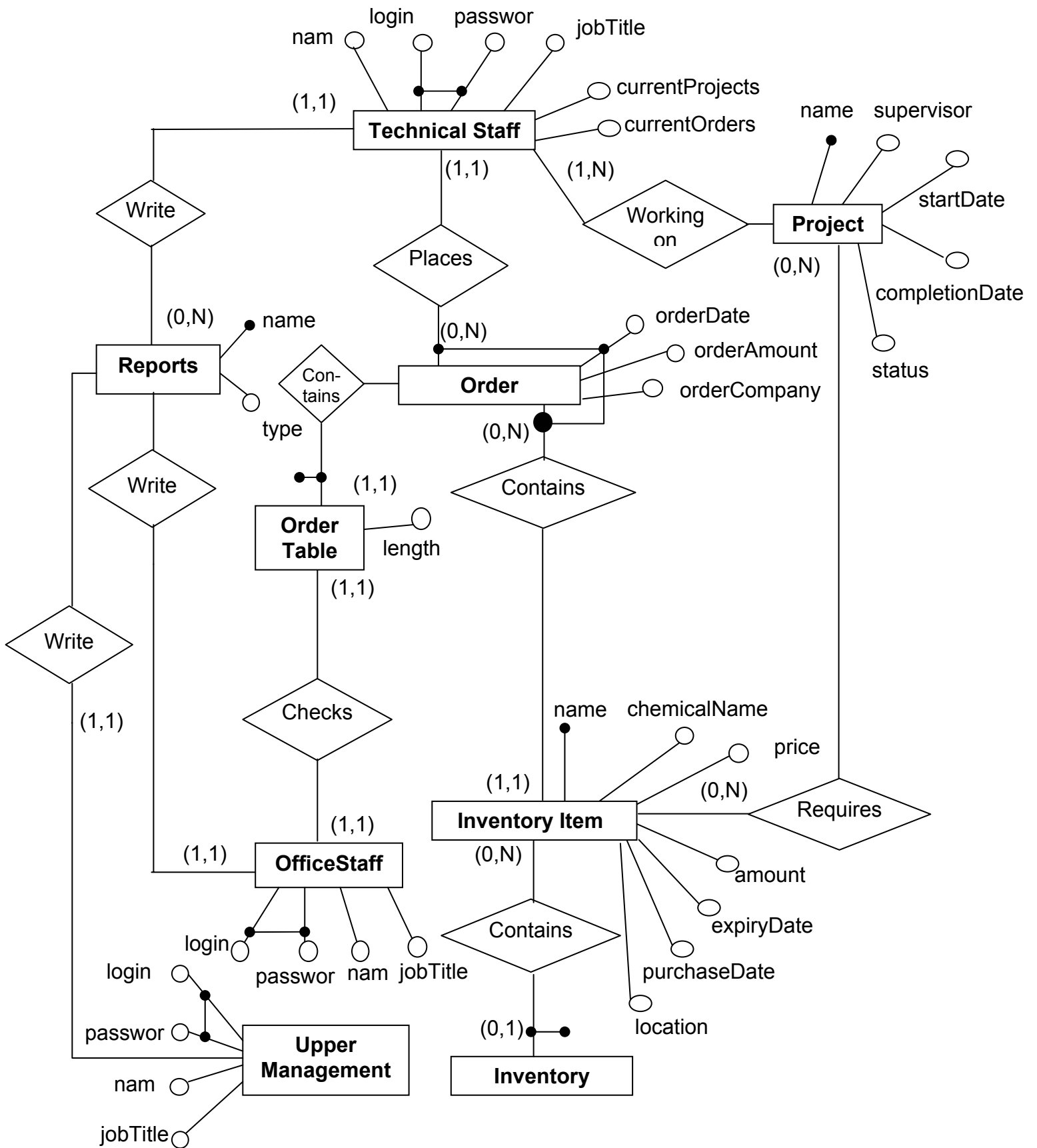
5. User Interface Design

The user interface design was very thoroughly considered since the success of this system relies very heavily on user acceptance. As mentioned previously, most users of this system will be novices and thus a simplistic and intuitive interface is mandatory. As such, we decided that each screen should present the user with as little information and as few choices as possible. In this manner, the users will not be inundated with data; they will simply be presented with a small number of choices that they can select from in order to get the information they need. This fulfills the non-functional user interface requirements as laid out in section V (see Appendix 6).

Our list of functional requirements specifies a list of data we are required to be able to accept from the user, or display to the user. The following is a tree of screens and functions that our interface will incorporate in order to facilitate this input/output (I/O) of information. Items appearing in boxes, written in regular print are screens that a user can navigate to i.e. there will be a 'Login' screen, and a 'Main Options' screen. Items written in italics will not appear on their own screen, rather they will be accessible through buttons present on the screen that points to them i.e. the 'remove item' function will be accessible through a button on the 'inventory' screen. The numbers after the screen or function indicate the functional requirement the information they contain will fulfill (see Appendix 5). The connections between boxes illustrate the path a user would have to take through the system in order to obtain that data or execute that function. Items that are in bold are illustrated in Appendix 28. Of course what follows is not a comprehensive list of all possible interactions with the system, but it provides an overview of the major functions that will be available. See Appendices 27 and 28 for a full discussion of the user interface design.

```
                        ┌─────────────┐
                        │   Login     │
                        └─────────────┘
                               │
                               ▼
                        ┌─────────────┐
                        │ Main Options│
                        │   Screen    │
                        └─────────────┘
```

Login

Main Options Screen

Search Tab   **Projects Tab**   **Inventory**   **Scientists**   **Orders Tab**

Print Reports
Iij1-3

Inventory List
Ia1-6, IIa, IVa1i, ii

Add New Item
IIb

Place Order
IIh

Print Order Screen
Ili

Check Out Item
IIf

*Remove Item IIc
Check Availability IId
Check Location IIe*

Update Item
IIg

Cancel Order

Project Info
Ic1-4, IVa1iii,
IVa2i, Ivc1-2

Add Project

Scientist Info
Ib1-2, IVa1iv,
IVa2ii, Ivb1-3

Add Scientist

21

Conclusions

In light of the above analysis, we feel that these design parameters will be sufficient to create a fully functional, easy to use and very powerful inventory tool for PRO. We have mentioned virtually every requirement by number in this document and have dealt with all others, not specifically mentioned here, previously in the Requirements Analysis. This system design proves that every requirement can be met or exceeded by following the recommendations presented. Thus we feel that our solution is still the most viable, feasible and appropriate solution to the inventory problems at PRO.

Appendices

Appendix 1

Brainstorming Session 1 – March 27th 2001

Notes: First team meeting. Preliminary project ideas.

Summary:

This meeting was just a very informal discussion on how we would proceed with the System Design phase of the project. We decided to continue using Process Research Ortech (PRO) as our subject company. We felt that our Feasibility Study and our Requirements Analysis contained enough information for us to proceed with the design phase, so no further information gathering activities were scheduled. We did not preclude the idea of conducting further interviews etc., but at this point we did not see the need. All pertinent information outlined in any of the previous studies was compiled and appears here in the next few appendices.

Appendix 2

Explanation of Appendices – March 27th 2001

The following appendices (Appendix 3 – Appendix 6) were originally created for our Feasibility Study or Requirements Analysis. They include a company structure diagram, a description of the current system and lists of requirements for the new system. These documents are included as reference material only. Any additional information that is required can be found in either the Feasibility Study or the Requirements Analysis. All the borrowed appendices have their original date of creation noted in their respective "Notes" section.

Appendix 3

Organizational Chart – March 27th 2001

Notes: Organizational hierarchy at PRO.   This is included to impart and understanding of the company we are designing this system for. *Document originally created March 1st 2001 for Requirements Analysis.*

Board of Directors
The board manages the direction of the company, in terms of what new projects should be undertaken and what venues of research PRO should pursue.

President
The president's job is to ensure the smooth working of the company and the completion of projects on time and to oversee the managers. The president also interacts with outside consultants, who are specialists in different lines of work who offer their expertise on certain projects.

Consultants
Consultants offer their expertise when such expertise cannot be found in-house at PRO.

Office Manager
The office manager ensures office tasks such as payroll are done properly. She is also in charge of procuring inventory and other supplies for the smoothing functioning of PRO.

Manager
The Manager oversees the day to day operations of the company he also supervises the chemists, technologists, metallurgists and other scientists

Technologists
Technologists carry out pilot plant and lab scale experiments.

Chemists
Chemists carry out chemical processing such as precipitation, HPLC and other chemical analysis.

Metallurgists
They deal with various metallurgical processes in the projects PRO deals with.

Technician
Technician reports to any one above him, depending on project load and where help is needed.

Hydro-Metallurgists
Hydro-Metallurgists deal with various hydro-metallurgical processes such as froth-flotation etc.

Appendix 4

PRO Inventory System Flow Chart – March 27th 2001

Notes: This document summarizes the information we have obtained concerning the current inventory system at PRO. *Document originally created February 26th 2001 for the Requirements Analysis.*
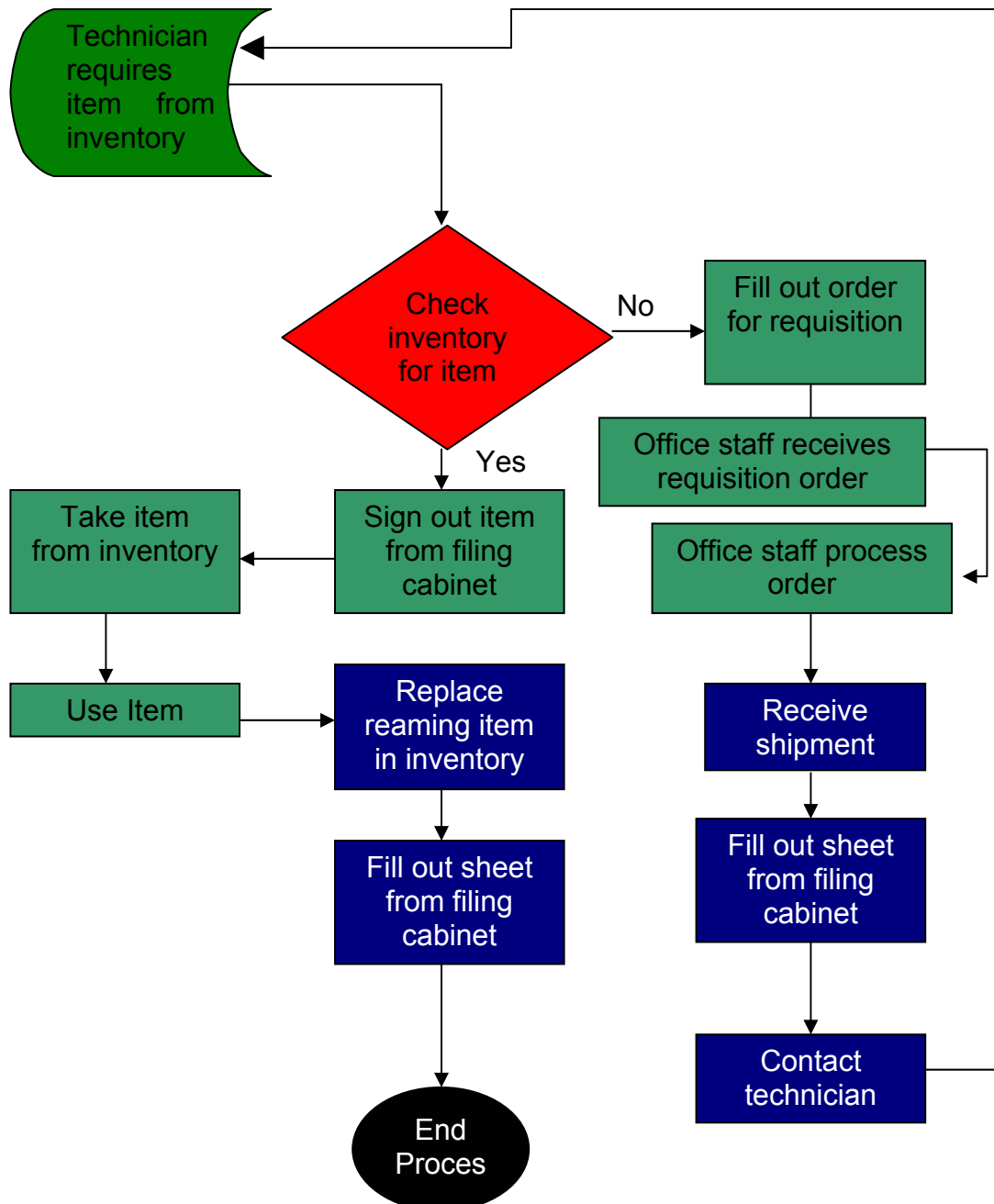
Starting with a scientist who is working on a project, the system works as follows. When a scientist needs an item for a project, they go to the filing cabinets to search for the file that pertains to the item they require. That item will be filed under one of the possibly multiple accepted names for the material. Once the file is located, the scientist will look inside to see how much of the substance is left in inventory. Previous users should have recorded how much was left after they put the item back in inventory. If there is enough of the item for them to use, they will sign their name in the file, the date and how much of the item they are taking. They replace the file, return to their lab, use the item and return it back to inventory if there is any left. If there is some left, they sign the item back into inventory using the same file as before. They write their name, the date and the amount they are returning. After a quick calculation they record how much of the item in total is left in inventory. After recording the information in the file, they replace the file and the process ends.

If they do not find the file, they look in the inventory room and simply take the item without recording anything. If there is not enough of the item, or there isn't any of the item at all, they fill out an order form. They put their name, the project name (and supervisor) on the form as well as the item name and how much they are requesting. They hand the order form off to an office worker who will give it to the Office Manager. The Office Manager will look at all the orders she has received and look through supplier's catalogues to find the best prices. Once she has located the best deals, she calls up the supplying companies and places the orders. In a few days, the items arrive. Once they are placed in the office, the Office Manager or one of the office staff go through the inventory files and update all the files that correspond to materials that have just been delivered. The person will record the date, the amount of substance delivered, and calculate how much there is in total now. They then replace the files and call the scientists who ordered the items (their names are on the order forms). The scientists pick the items up from the office and take them back to their labs. They use what they need and then check the remainder back into inventory. As above, they write down the date, and how much they are returning. A calculator tells them how much is left in total now in inventory. That number is recorded, the file is replaced and the process ends.

Appendix 5

Functional Requirements – March 27th 2001

Notes: List of functional requirements for the proposed inventory system with accompanying explanations. *Document originally created March 1st 2001 for the Requirements Analysis.*

- **I Details of data storage**
  - a) Inventory Items
    - 1. Name
    - 2. Location
    - 3. Usage
      - i) Dates of usage
      - ii) Projects usage
      - iii) Personnel usage
      - iv) Amounts of usage
    - 4. Date of Order
    - 5. Expiry date
    - 6. Cost of item
  - b) Scientists names
    - 1. Projects they are working on
    - 2. Current orders they have placed
  - c) Projects names
    - 1. Materials needed
    - 2. Project start date
    - 3. Project finish date
    - 4. Supervisor
- **II Details of output**
  - a) Inventory list screen
  - b) Add new item screen
  - c) Remove item screen
  - d) Check availability screen
  - e) Check location screen
  - f) Check-out item screen
  - g) Update item screen
  - h) Place order screen
  - i) Print order screen
  - j) Print reports screen/Print hard copy
    - 1. Inventory Reports (various reports)
    - 2. Personnel Reports (various reports)
    - 3. Project Reports   (various reports)

- **III Details of input**
  - a) Paper documents
    - 1. Current inventory system info (paper files)
    - 2. Shipping order
      - i) Item name
      - ii) Item amount
      - iii) Item date
      - iv) Persons name who ordered the item
- **IV Details of Information Processing**
  - a) Inventory item
    - 1. Usage details
      - i) Amount remaining
      - ii) Time to expiry
      - iii) Amount used by project
      - iv) Amount used by Personnel
      - v) Amount used MTD (month to date)
      - vi) Amount used YTD (year to date)
    - 2. Cost details
      - i) Dollar amount used by project
      - ii) Dollar amount used by personnel
      - iii) Dollar amount of item used MTD
      - iv) Dollar amount of item used YTD
  - b) Personnel usage
    - 1. Dollar amount of items used YTD
    - 2. Dollar amount of items used MTD
    - 3. Dollar amount of items used per project
  - c) Project usage
    - 1. Dollar amount of items used YTD
    - 2. Dollar amount of items used MTD

I. Details of data storage

The major responsibility of the system will be to hold various types of data. a) It will need to store inventory data for all inventory items such as the item name, where it's being used, who is using it, where it is currently, when it was ordered, when it will expire (if applicable) and how much it costs.
b) The system will also need to store information about the technical staff i.e. the names of the supervisors, technicians and technologists. It will need to associate those names with the projects they are working on and the orders they have placed.
c) Finally, the system will have to keep track of all the projects that are going on. Project start and finish dates will need to be recorded as well as the names of the staff members working on it (supervisor at least). It will need to relate these projects to the inventory materials they require.

II. Details of output

The outputs of the system will usually be screens, but the ability to print any report will also be present.
a) This screen will allow the user view a list of all or some of the items in the inventory list.
b) When a new item arrives that has never been used in the company before, the office staff will use this screen to create a new entry in the database for it.
c) If an item is no longer used by the company, it can be removed from the database with this screen (e.g. it is found to be toxic etc.).
d) This screen allows the user to find out how much of a material is left in inventory.
e) A user can use this screen to check the location of an item in the building. For example, if a canister of gas must be kept outside the inventory room in a shielded room, the user can request that room number here.
f) Scientists will use this screen when they remove an item from inventory. They will record their name, what they are taking, how much, etc.
g) This screen will be used when items are being checked back into inventory by scientists, or by office staff to update the database when ordered items arrive.
h) Scientists will use this screen to place orders for materials.
i) Office staff will use this screen to print a list of all unfilled orders placed that day, or hour, or week, etc.
j) Any staff member can print reports about inventory items, personnel or projects.   Inventory reports include information about all or some inventory items i.e. where they are, how much is left, who ordered it, when it was ordered, when it will expire, etc. Personnel reports will contain information on the projects people are working on i.e. project

names, what materials they are using, how much the person has spent so far, etc. Project reports give much the same information as personnel reports. For given project names, they return the cost of the project, the materials it uses, its start and finish dates, etc.

III. Details of input

There are not many inputs at all into the system from anything other than computerized user updates. Very few paper documents, no telephone calls or outside system updates will input information into the system.

I. The current files have to be input into the system, but once that is completed, it will never have to be done again. When orders arrive, office staff will have to enter most of the data on the shipping forms into the database; this will be a reoccurring task.

IV. Details of Information Processing

The system will have to do a lot of information processing and internal calculation. That is one of the main purposes of the system in the first place i.e. to automate tedious chores.

a. The system will have to take the user input data concerning every inventory item and perform a series of calculations. It will have to calculate how much of the item is left, how long until it expires, how much is being used by specific people, how much is being used by specific projects, etc. It will also have to perform monetary calculations concerning the dollar amount of those items used by specific people, specific projects, etc.
b. Independent of specific inventory items, the system will need to calculate how much money specific people are spending in total.
c. Similarly, the system has to calculate how much money specific projects are using as well.

Appendix 6

Non-functional Requirements – March 27th 2001

Notes: List of non-functional requirements for proposed inventory system with descriptions of PRO's ability to satisfy them. *Document originally created March 1st 2001 for Requirements Analysis.*

- **I Software Requirements**
    - a) MS Windows 2000 Server
    - b) MS Access
    - c) MS C++
- **II Hardware Requirements**
    - a) Server
        - 1. Pentium-II 400 MHz
        - 2. 300 MB hard disk space
        - 3. 256 MB RAM
        - 4. 10Base-T Network Interface Card
        - 5. Laser Printer
    - b) Clients
        - 1. Pentium 100 MHz
        - 2. 64 MB RAM
        - 3. 10Base-T Network Interface Card
- **III Security Requirements**
    - a) User name and Password identification for all users
    - b) MS Windows Primary Domain Controller
- **IV Reliability/Survivability Requirements**
    - a) Monday to Friday availability
    - b) 8:00am – 6:00pm availability on those days
    - c) Data restoration within 24 hours of data loss
    - d) Daily/Nightly back-ups of database
- **V Interface Requirements**
    - a) Simple interface
    - b) No large user manual required
    - c) Short training session
- **VI Lifecycle Requirements**
    - a) System upgradeable
    - b) Development time < 6 months
- **VII Economic Requirements**
    - a) Approximately $50, 000 development cost
        - 1) Salaries
        - 2) Software
        - 3) Hardware
        - 4) Installation

1. Software Requirements

The inventory system will run on a Microsoft Windows 2000 Server platform. Microsoft Access will drive the database. Microsoft C++ (Microsoft Foundation Classes) is required to develop the GUI (Graphical User Interface) application that interfaces with the database.

PRO already possesses licenses for Win2000 server and MS Access and is currently using both pieces of software. Since a development team will need MS Visual C++ to develop a GUI application, a license for MS Visual C++ will need to be obtained.

2. Hardware Requirements

A Pentium-II 500 MHz CPU would be ideal to run the proposed system. In terms of hard disk space, an Access database system with an upper limit of 1000 items (with about 10 fields each), would take about 300 Megabytes (absolute maximum). In terms of RAM, about 256 Megabytes will ensure fast search times, and fast query retrievals. Since this information system is accessible through the network, a 10Base-T Network Interface Card is required. Finally, a printer will be required to print out hardcopy reports.

The network server used at PRO already meets or exceeds these specifications. PRO possesses many other computers that basically fit this description as well. With a few minor upgrades these computers can easily satisfy these requirements. These extra machines can be set up as workstations around to company to act as access points for the database. PRO already has a laser printer in the office that can be used for printing reports off the system.

3. Security Requirements

Access to the system will be granted through entry of a login name and password authenticated by a Microsoft Windows NT based Primary Domain Controller. Users will belong to different groups based on their login names, such as the technical staff group, office staff group or upper management.

PRO already has Windows NT based Primary Domain Controllers, which can authenticate passwords. This technology is currently in use at PRO.

4. Reliability/Survivability Requirements

High reliability is of paramount importance, so a stable operating system and database is required. This system needs to be available at least 5 days a week (Monday – Friday), between 8am and 6pm. It requires daily back-ups to guard

against accidental data loss. Data need to be restored within 24 hours to ensure smooth operation of the company.

PRO already has a daily data backup system, so no new system needs to be implemented. A larger data backup media needs to be used backup the database though i.e. Zip Disk, or re-write able CD's.

5. Interface Requirements

The employees at PRO are not extremely comfortable with computers. They are all computer literate, but would reject a complex interface. Thus the interface needs to be simple (point and click), with lots of graphics as opposed to text. The usage of the system should be intuitive to negate the need to read large user manuals. The employees could only spare a few hours in a single day to be trained on the system. There are a small number of them, so a single training session is viable, but the system has to be simple enough for them to grasp it in those few hours.

The system is being designed with very a very simple point and click interface and will use screens that closely resemble the paper documents they use now. We are confident that any person can be taught to use this system in a minimum amount of time.

6. Lifecycle Requirements

The system should be upgradeable so it can match the needs of the company as it changes. Development time should also be shorter than 6 months as this is the time limit the company set for the project.

The programming languages and platform selected for this project (see software requirements) are flexible enough to allow easy upgrades to be developed for the foreseeable future. Development time for this project has been estimated at 4 months (see Appendix 4)

7. Economic Requirements

PRO has set an approximate upper limit of $50,000 dollars for the development of this project. This figure must cover the full cost of production including developer salaries, software licenses, applicable hardware and installation of the system.

This system will cost $58, 700 in the first year to develop, install and maintain. The system will pay for itself after the 18th month and will turn a profit of $38, 819 after only 36 months (with cumulative benefits to that point in excess of $100, 000)

Appendix 7

Brainstorming Session 2 – March 29th 2001

Notes: Further information gathering planned.

Summary:
      We determined that the information we currently had was not detailed enough to allow us to do a proper design. We planned another trip to PRO to conduct our own surveillance of their computing capabilities and to perform random interviews. We required detailed information on the number of computers they had, where they were located, the capabilities of each machine, the type of network they had, software licenses they owned etc. The interviews were required for us to determine employee feelings on changing hardware, changing software platforms etc. We felt this information was critical, so we arranged a trip for the next day.

Appendix 8

Information Gathering Summary – March 30th 2001

Notes: Compilation of all the information obtained on PRO's computing capabilities.

All hardware, software and network details are summarized in Appendix 9 in the Global Architecture Section.

After conducting random interviews with office staff members, management and scientists, we determined a few facts about the average employee at PRO. First of all, most PRO employees are not very comfortable with computers. Currently, computers are used for e-mail and printing documents; a few employees use them for Internet browsing and file sharing. Most scientists use computers in their research to either record data or do analyses; most of the data entry is handled by technicians. As a result of this casual use of computers, many employees have not learned how computers work. They simply memorize the procedures required to do their usual tasks and leave it at that. Most of them realize that they are very dependant on the "look" of the applications and the operating system (OS). They tell us that if the buttons and menus changed locations, they would be hard pressed to perform all the tasks they are used to doing. We asked them a few general questions about the machines they currently used. Everyone we asked was aware that their computer ran Microsoft (MS) Windows, and most of them knew that MS Outlook was the program they used to check their e-mail. When asked about other operating systems like Linux and Unix, very few people understood what we meant. Only 2 people that we talked to had actually ever worked on a Unix machine.

Appendix 9

Current Computing Capabilities – March 30th 2001

Notes: Description of the current state of PRO in terms of hardware, networking and software capabilities.

1. Server Computer - (Glass box)

    a. **Hardware**
        i. Pentium II @ 400 MHz.
        ii. 256 MB DRAM
        iii. 8 Gig HDD (currently 6 Gig free space)
        iv. 8 Gig Tape backup
        v. 10/100-BaseT NICs
        vi. 32X CD Reader
        vii. 4X CD Writer
        viii. 1.44 FDD
    b. **Software**
        i. Microsoft Windows 2000 Server
        ii. Microsoft Office 2000 Professional
            1. Access
            2. Word
            3. Excel
            4. Outlook
            5. PowerPoint

2. Client Computer(s) - (Glass box)
(Technical Staff, Office Staff or Upper Management machines)

    a. **Hardware**
        i. Pentium @ (100-200) MHz.
        ii. 32-64 MB RAM
        iii. 2-4 Gig HDD
        iv. 12-32X CD Reader
        v. 10/100-BaseT Network Interface Cards (NIC)
    b. **Software**
        i. Microsoft Windows 2000 Professional
        ii. Microsoft Office Professional
            1. Access
            2. Word
            3. Excel
            4. Outlook
            5. PowerPoint

3. Networking Components

    a. **Hardware**
       i. 100 Base-T twisted pair Cat. 5 cable (copper)
      ii. 32 port, switched hub
     iii. HP LaserJet 5P *Printer* & Jet-Direct LAN interface
    b. **Software**
       i. NetBEUI Protocol
      ii. TCP/IP Protocol
     iii. Windows Primary Domain Controller

4. Software Licenses

    a. Microsoft Windows 2000 Server (2 Licenses)
    b. Microsoft Windows 98 (Multi-site license)
    c. Microsoft Windows 2000 Pro (5 Licenses)
    d. Microsoft Office 2000 Pro (Multi-site Licenses)

Floor Plan for Process Research ORTECH
Not to accurate scale.

Legend

Technical Staff

Upper Management

Office Staff

Switched Hub

Windows™ 200 Server

Pilot Plant Floor

Inventory Room

30 feet

Offic

Offic

Recepti

Labs

Labs

Labs

10'

Offic

Offic

Offic

Labs

100 feet

40
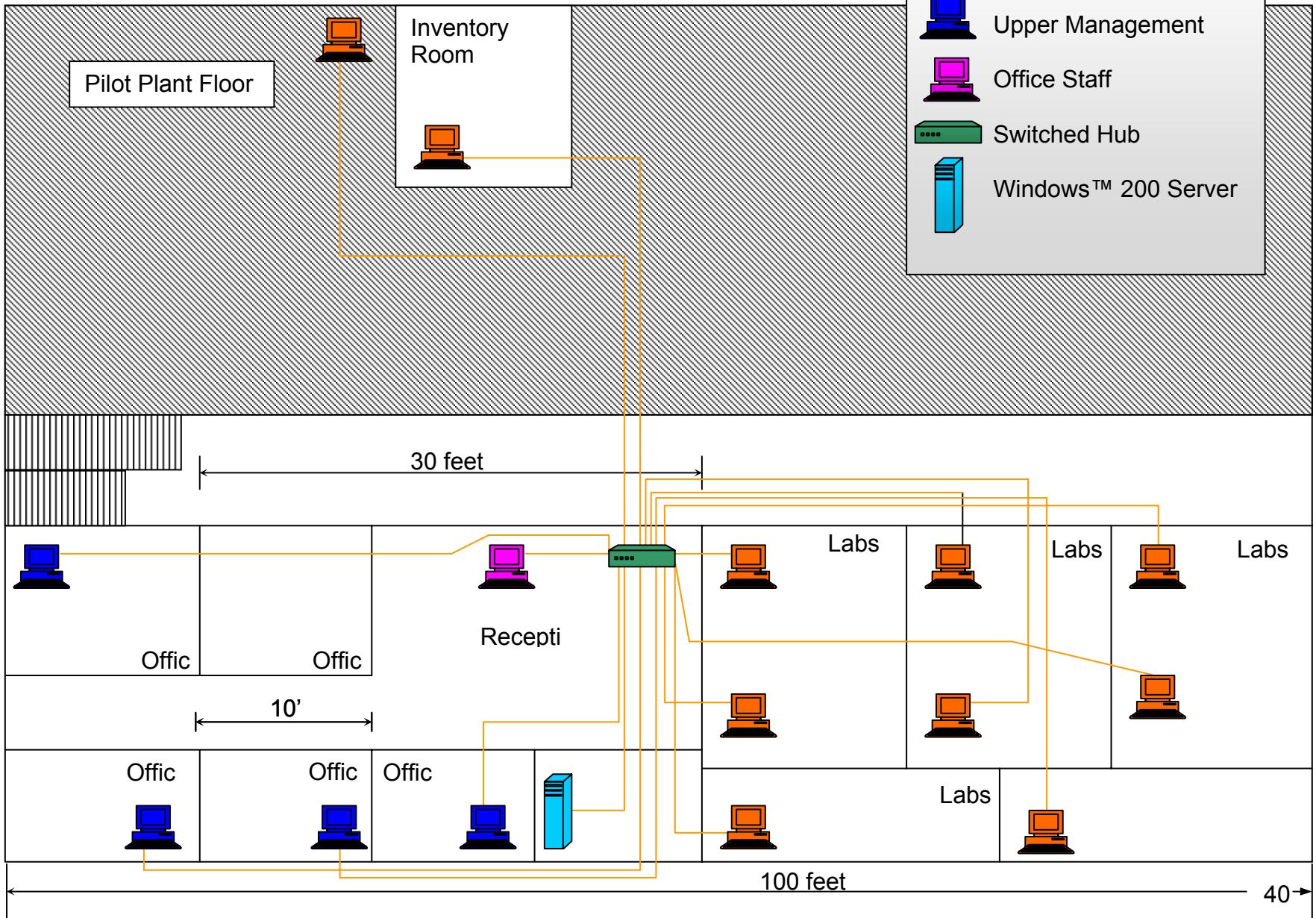
Appendix 10

Hardware Considerations – April 1st 2001

Notes: Analysis of the current situation at PRO, and proposed solutions with regards to computer hardware for the new inventory system.

Summary: Given economic and human factors, we decided to maintain the current hardware setup.

The current hardware situation at PRO is actually quite good (see Appendix 9). The machines are relatively new and more than adequate to handle the simple database solution we are proposing. The machines are all microcomputers, very open and very glass box. Hardware requirements for the new system can be found in the non-functional requirements, part II (see Appendix 6). It is clear to see that the current hardware at PRO matches or exceeds these specifications. The current server runs at 400 MHz, and has more than enough RAM and free hard disk space for the new system. Each client machine has adequate processor power, RAM and hard disk space to use the new system as well. Thus simply maintaining PRO's computer infrastructure as it currently stands fulfills non-functional requirements IIa1, 2 and 3 as well as requirements IIb1 and 2. With 15 client machines, there are more than enough terminals for everyone to access one. They are present in all the offices, labs and work areas, thus there is no need to purchase more. With nightly 8 Gig tape backups, the database can be stored every night and retrieved with little or no hassle in case of data loss (non-functional requirements IVc and d).

We considered upgrading to proprietary machines, but we felt that the cost PRO would incur would be unnecessary (see non-functional requirement VIIa3). Economics was also the reason we decided against upgrading the client machines to workstations or the server to a minicomputer. PRO is a small company, so they can't afford to pay for the actual machines, or for the technical assistance that is required to service and maintain them (usually only available from the vendor). With open, glass box systems, any technician can come in and maintain or upgrade the computers thus aiding in fulfilling reliability and upgrade requirements IVa, and b and VIa. In addition, by maintaining the current hardware, we simplify the bug fixing procedure when we install the system. All the current hardware has been in place for at least two years, so if problems occur, we can attribute them to the new software.

Probably the most significant reason to maintain the current hardware setup is the human factor. As mentioned previously, most of the employees at PRO are computer novices and are very comfortable with the machines they have now (see Appendix 8). Replacing the hardware would require them to learn how to use a new machine. New machines could potentially be very different from the ones they have been using up till now (especially in the case of some

proprietary systems). We feel that user resistance is to be avoided at all costs, thus there is strong support for maintaining the current hardware.

Appendix 11

Network Considerations – April 1st 2001

Notes: Analysis of the current situation at PRO, and proposed solutions with regards to networking for the new inventory system.

Summary: Given economic and human factors, we decided to maintain the current network architecture.

The current network at PRO is set up as a simple peer-to-peer, TCP/IP Windows™ network. Currently, it is mainly used for email, the Internet, printing and file sharing (see Appendix 9). The cables are copper Ethernet, capable of transmitting at 100Mbps. The hub supports 32 users and 100Mbps, thus there is more than enough room and bandwidth for all 15 machines to be accessing the system at once. The machines are an acceptable distance apart, and the workload can be split and efficiently channeled by the switched hub. The new system can use the exiting network to connect to the database (using TCP/IP), query the database and return results. In addition, there is a network printer (an HP Laser Jet) accessible from anywhere in the LAN. Thus non-functional requirements IId and e and VIIa3 are satisfied by the current setup with no need to upgrade (see Appendix 6). The current network has an acceptable amount of downtime and since it has been in use for 2 years, we know it is reliable (non-functional requirements IVa and b).

There is no need to consider upgrading to a WAN since remote access is not a function of the new system. Upgrading to fiber-optic cables was considered, but rejected on the basis of cost. Token ring and ATM architectures were rejected on the basis of being too inefficient and too complex respectively.

Again, one of the main reasons to keep the current setup is the human factor. Networks can be very daunting for novice computer users. Most of the employees at PRO don't understand how the network operates, they simply memorize procedures for network printing and file sharing etc. Changing the network architecture may change those procedures and again elicit strong user resistance. Thus the decision was made to maintain the current network architecture.

Appendix 12

Software Platform Considerations – April 1ˢᵗ 2001

Notes: Analysis of the current situation at PRO, and proposed solutions with regards to the software platform for the new inventory system.

Summary: Given reliability and lifecycle requirements, we decided to maintain the current software system at PRO. The only additional tool required for the development of the new system is MS C++ foundation classes to design the user interface.

      The current software situation at PRO is very impressive (see Appendix 9). They have Windows 200 Server running on their server and Windows 2000 professional running on every client machine. They have MS Office Professional on every machine, which includes Access (the database program which we will use to construct our inventory database). This fulfills non-functional requirements Ia and b (see Appendix 6). The only additional piece of software required is MS C++ which will be used to design the user interface. A license and a copy of the software can be obtained form any one of a number of authorized Microsoft distributors for about $100 dollars, which is still well within the economic limits set out by section VII in the non-functional requirements (see VIIa3). Microsoft software is very widely used, and there is a plethora of documentation and cheap technical support for it. Also, it is relatively safe to assume that Microsoft will be in business for some time to come. Thus we can be sure of continued technical support, and frequent software upgrades (non-functional requirement VIa).

      We looked into designing our database with Oracle, DB2 or MSSQL, but all of these are more expensive, harder to install and troubleshoot than Access. PRO already has licenses for Access and employees are familiar with the "look and feel" of Office Suite products. Again, human factors turn up as being important here. PRO employees are comfortable with Windows and MS products in general. As before, many of them have simply memorized the procedures required to check their e-mail or print documents. Changing operating systems or presenting them with unfamiliar software styles would elicit user resistance. Thus the decision was made to use the software currently available at PRO.

Appendix 13

Global System Architecture Conclusions – April 3rd 2001

Notes: A compilation of the results from the preceding analyses

Items in italics need to be purchased. Items in regular script are already present at PRO.

Hardware

    a. Server – Microcomputer, open, glass box.
    b. Pentium II @ 400 MHz.
    c. 256 MB DRAM
    d. 8 Gig HDD (currently 6 Gig free space)
    e. 8 Gig Tape backup
    f. 10/100-BaseT NICs
    g. 32X CD Reader
    h. 4X CD Writer
    i. 1.44 FDD

Clients – Microcomputers, open, glass box

    a. Pentium @ (100-200) MHz.
    b. 32-64 MB RAM
    c. 2-4 Gig HDD
    d. 12-32X CD Reader
    e. 10/100-BaseT Network Interface Cards (NIC)

Network – LAN

    a. 100 Base-T twisted pair Cat. 5 cable (copper)
    b. 32 port, switched hub
    c. HP LaserJet 5P Printer & Jet-Direct LAN interface

Software – MS Windows and MS Applications

    a. Microsoft Windows 2000 Server
    b. Microsoft Office 2000 Professional
    c. Microsoft Windows 2000 Professional
    d. NetBEUI Protocol
    e. TCP/IP Protocol
    f. Windows Primary Domain Controller
    g. *Microsoft C++ Foundation Classes (must be purchased)*

Appendix 14

Explanation of Diagrams – April 5th 2001

After gathering and summarizing all the information we received from PRO, we set about trying to model the new inventory system. We adapted the class diagram from the Requirements Analysis (which appears here in Appendix 15) and began to model the functions of specific classes in state or activity diagrams; we also modeled the system as a whole with an E-R diagram. Diagrams for some classes were trivial; others were more complex. We have included in the next few appendices all the diagrams we thought were necessary to provide a complete understanding of the workings of each class. Trivial diagrams have been omitted for the sake of brevity. Classes describing human actors were usually modeled in Activity Diagrams; classes representing internal objects were usually modeled in State Diagrams. Each diagram comes with a textual description. If required, use case diagrams can be found in the Requirements Analysis.

Appendix 15

Class Diagram for Inventory System – April 5th 2001

Notes: Class diagram follows after text.

Our design for the new system is modeled in the class diagram. All users, objects and interactions are presented in some form on the diagram. Class descriptions are as follows:

Class Staff Member

Staff member is a parent class with three subclasses (office staff, technical staff and upper management). All of the staff members have names, logins, passwords and job titles, all of which the system will keep track of. Staff members can obviously log onto the system and use it. The addEmployee function simply generates a new login and password for an employee and tags them as being an office staff member, a technical staff member or a member of upper management. Staff members can change their password if they desire, but not their login. Every staff member can print every kind of report, both on the screen and on hardcopy. Each staff member can create multiple reports.

Class Office Staff

Office staffers can be either the Office Manager, or general office staff. They have no special attributes of their own (they just inherit from the Staff Manager class). The office staff is responsible for reading the orders from the order table, and dispatching them to the various suppliers. There is only 1 order table on the system and it is dealt with by only 1 office staff member (usually the Office Manager).

Class Technical Staff

Technical staff can be project supervisors, technicians or technologists (job title). They have attributes over and above those of a regular staff member. They have various projects they are associated with and orders they have placed. Technical staffers are responsible for picking up items once they're delivered, adding their new projects to the system, and updating the system when they change projects. They can also be added and removed from the system. This is different from the addEmployee in the staff member class in the sense that scientist names added through the method in the Technical Staff class actually become items in the database. Scientist names are just like item names, they can appear on reports, be associated with other items in the system, etc. The names of other employees are never used in the system other than to identify them when they log in. Tech. staffers are associated with 0 or more

projects, and 0 or more orders in the system. They are pictured as having strong ownership over their orders (aggregation relation). This is because their name is associated with the order and they will be called when the order arrives.

Class Upper Management

These are the people on the Board of Directors, the President of the company and the Scientific Manager (oversees all the projects, and scientists). They have no attributes beyond those of a staff member and they have no operations beyond that of a staff member either.

Class Reports

Any staff member can generate reports. There are multiple types of reports and a lot of different kinds of data that can be displayed. Each report has a name and type. A report is owned by a single staff member, although a single staff member can have many reports. The Report class is responsible for the printing of reports on the screen and the generation of hard copy reports.

Class Order Table

The order table is constantly being updated as scientists place orders and office staff dispatch them. There is only 1 table in the system and it holds 0 or more orders. The Order Table class is responsible for combining orders for the same items, generating a virtual copy of the table for the office staff to view, adding new orders to the table and checking the length of the table. There will be a threshold value entered by the Office Manager that represents the maximum allowable orders that are allowed on the table.

Class Order

An order is placed by a single scientist (tech. staff). They list dates, amounts and supply companies. The Order class is responsible for creating the orders and canceling them upon request from a staff member. They are associated with that single staff member, and a single item (cannot order multiple items on a single order form). Orders are pictured as having an aggregation relationship with the order table and their order item. Obviously without orders, the order table would become empty and inventory items would run out (although both these items would still remain "alive" in the system).

Class Inventory

The inventory class represents the list of all the inventory items in the system. Fittingly, it has as its only attribute a list of all inventory items. It is associated with 0 or more of these items and has strong ownership over them (if the inventory disappeared, the items would only exist in the labs that were using

them). All staff members must search and update this inventory list when they are performing inventory operations.

Class Inventory Item

These are the individual items contained in the inventory list. They have names, prices, amounts, expiry dates, purchase dates and locations (i.e. a room number). They are associated with 0 or more projects, 1 order and 0 or 1 inventories. The Inventory Item class is responsible for checking the location of items, checking their availability, updating their status in the database (with regard to amounts, dates, etc.), adding new items to the list, removing items from the list, checking items out from inventory and changing project associations (when an item is required by a project or when it is no longer needed by a project).

Class Project

Projects are the whole reason for this system. They are the entities that require scientists to work on them, inventory items to complete and so on. Projects have names, supervisors (a tech. staff), a start date, a completion date, a list of items they require, a list of scientists who work on them and a status (nascent, in progress, halted, completed). They are associated with 1 or more technical staff (at least one to act as supervisor) and 0 or more inventory items (it's possible that they use no inventory items). The project class is responsible for updating the system when a new project is started, or when it is completed, updating the project item and scientist lists, and changing the status of the project.

StaffMember
- name
- login
- password
- jobTitle
- addEmployee()
- changePassword()

UpperMangement

Inventory
- listofItems

InventoryItem
- name
- chemicalName
- price
- amount
- expiryDate
- purchaseDate
- location
- projectAssociation
- checkLocationItem()
- checkAvailability()
- updateItem()
- addItem()
- removeItem()
- checkOutItem()
- changeProjectAssociation()

TechnicalStaff
- currentProjects
- currentOrders
- pickUpItem()
- addProject()
- removeScientist()
- addScientist()
- changeProjects()

OfficeStaff
- readOrderTable()
- dispatchOrders()

OrderTable
- length
- collateOrders()
- generateTable()
- addToTable()
- checkLength()

Order
- orderDate
- orderAmount
- orderCompany
- placeOrder()
- cancelOrder()

Reports
- name
- type
- printHardReport()
- generateReport()

Project
- name
- supervisor
- startDate
- completeDate
- itemList
- scientistList
- status
- startProject()
- completeProject()
- updateItemList()
- updateScientistList()
- updateStatus()

50

Appendix 16

Inventory Item State Diagram – April 7th 2001

Notes: State Diagram for the Inventory Item Class.

An inventory item comes into existence when it is added to the database with the addItem() function call in the Inventory Item class.

Idle
It enters the Idle state and remains idle until a technical staff invokes the changeProjectAssociation function passing this item's name as the argument. It will return to this state when an item formerly in use has its project association changed to 'none' and its status is updated with the updateItem() function.

In Use
It enters the In Use state when a user changes its project association to an active project and the item itself is available (verified with the checkAvailability() function). It will leave this state when its project association is changed to 'none'.

Depleted
It enters this state when a user changes its project association and it is unavailable (verified using the checkAvailability() function). It will leave this state and return to the idle state when its availability changes from 'unavailable' to 'available'.
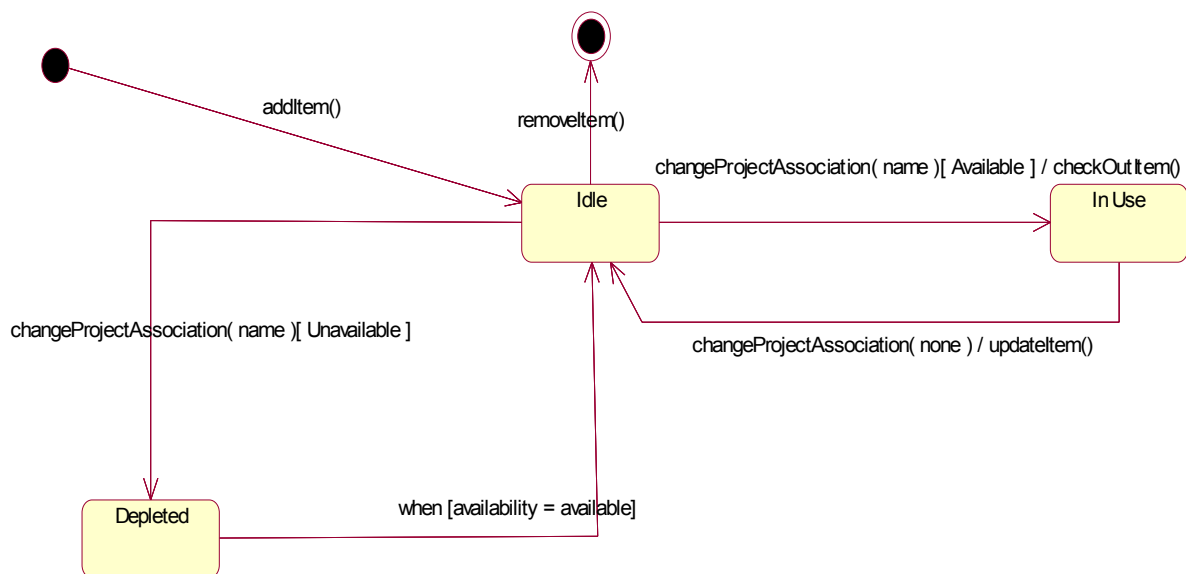
Appendix 17

Office Staff Activity Diagram – April 7th 2001

Notes: Activity diagram for the Office Staff class.

The activities listed here are only a subset of the possible activities an Office Staff member can perform. These activities illustrate only what an Office Staff member does in terms of dispatching the orders found on the Order Table.

Check Order Table Length – Office staffers must check the order table length in order to see if it has enough items to merit printing it and dispatching them.

Collate Orders – Once the office staff sees that there are enough orders, they will have the system combine orders for the same items.

Print Order Table – They will then prompt the system to print a readable version of the order table on the screen.

Dispatch Orders – Once the orders are on the screen they can dispatch them to the various suppliers.

Update Items – When the items arrive, they will update the database with the date, amount, price, etc. information.

Alert Scientists – After updating the database, they will alert the scientist that ordered the item so they can come and pick it up.
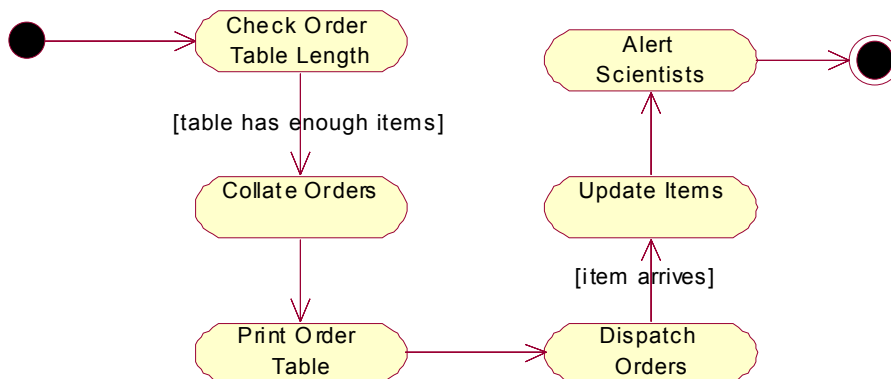
Appendix 18

Order Table State Diagram – April 7th 2001

Notes: State Diagram for the Order Table Class.

The Order Table will be created when the system is first started and will exist so long as the system is in operation. Thus there is no specific create or destroy method in the Order Table class as there would be in the order class. Upon creation, the table goes into the empty state.

Empty
It enters the empty state after creation, or after all the orders present have been dispatched. It will leave the empty state when a user adds an order to it.

Filling
The table enters this state when it contains 1 or more orders. It will remain in this state so long as the number of orders is less than the threshold value (that value is set by the office staff). It will leave that state when the number of orders is greater than the threshold.

Full
The table is full when it has more orders than the threshold value. The table will remain in the full state (possibly accumulating even more orders) until an office staff member reads the table (calls the generateTable() function) and empties it out.

destroy()

create()

Empty

readTable() / generateTable()

addToTable(order)

addToTable(order)[length <= threshold]

Filling

addToTable(order)[length > threshold]

Full
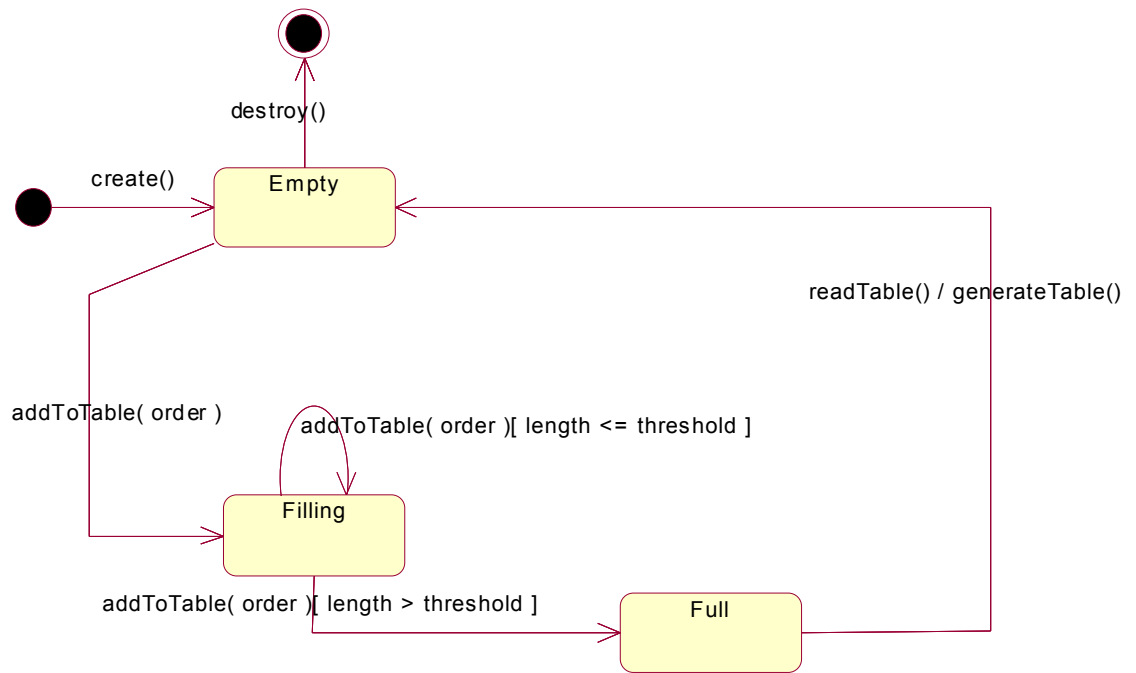
Appendix 19

Project State Diagram – April 7th 2001

Notes: State diagram for the project class.

A project comes into existence when the startProject() function is called. It immediately enters the nascent state.

Nascent
A project is nascent when it is first created and remains in this state until it begins gathering inventory items.

Halted
When a project tries to gather an item that is unavailable at any point in its progress, it enters the halted state. It will remain in this state until the item it requires is updated in the inventory list.

In Progress
If the project can gather all the items it requires, it enters the in progress state. It will remain in this state until the completeProject() function is called and the project finishes.
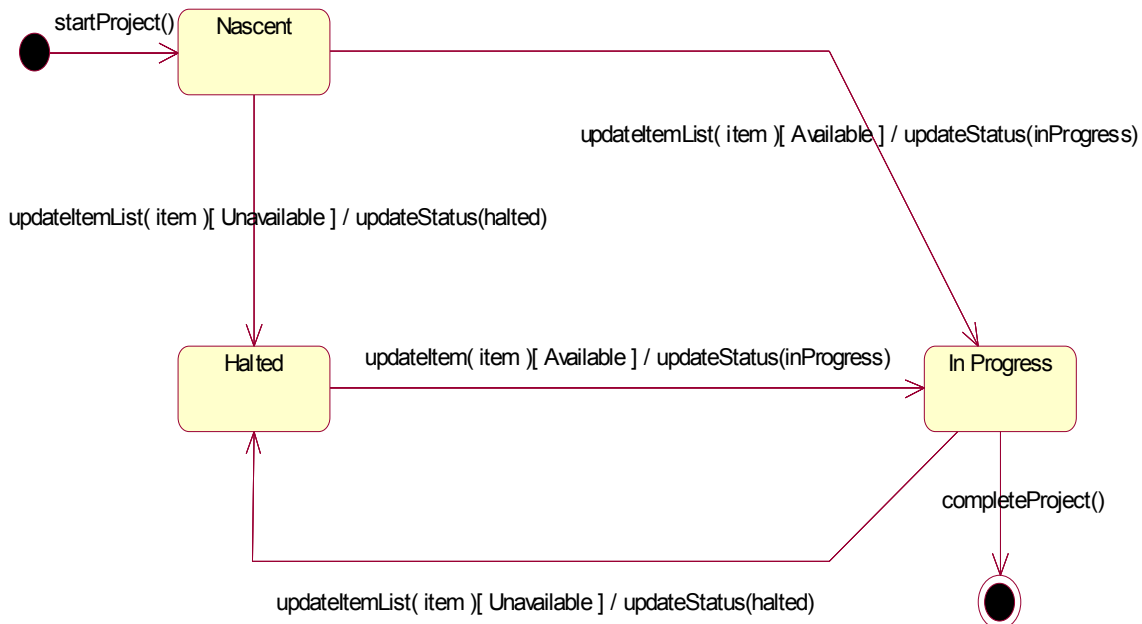
Appendix 20

Technical Staff Activity Diagram – April 7th 2001

Notes: Activity Diagram for the Technical Staff Class.

The activities listed here are only a subset of the possible activities a Technical Staff member can perform. These activities illustrate only what a Technical Staff member does in terms of project work and material gathering.

Sign up for project – Technical staff members can be assigned to a project in the database.

Obtain Materials – They can then go to the inventory list and get all the materials the project requires.

Place Order – If the item is not found, they can place an order for it.

Pick Up Item – When the item arrives, they can pick it up from the office.

Work on Project – Once all the items are obtained, they can begin work on the project.

Return all items to inventory – When the project is completed, they check all the items back into inventory.
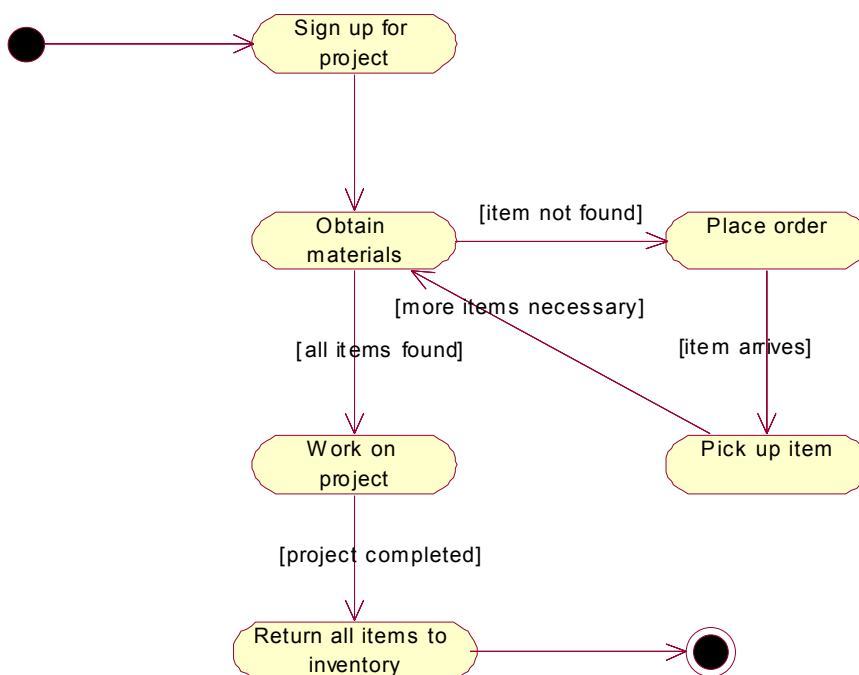
Appendix 21

ER Diagram – April 8th 2001

Notes: Preliminary E-R Diagram for the Inventory System.

This ER Diagram follows directly from the class diagram (see Appendix 15). A description of the entities and relationships depicted here can be discerned from the description of the classes there. Note that a data dictionary and a list of business rules follows the diagram.

(next page).

Data Dictionary

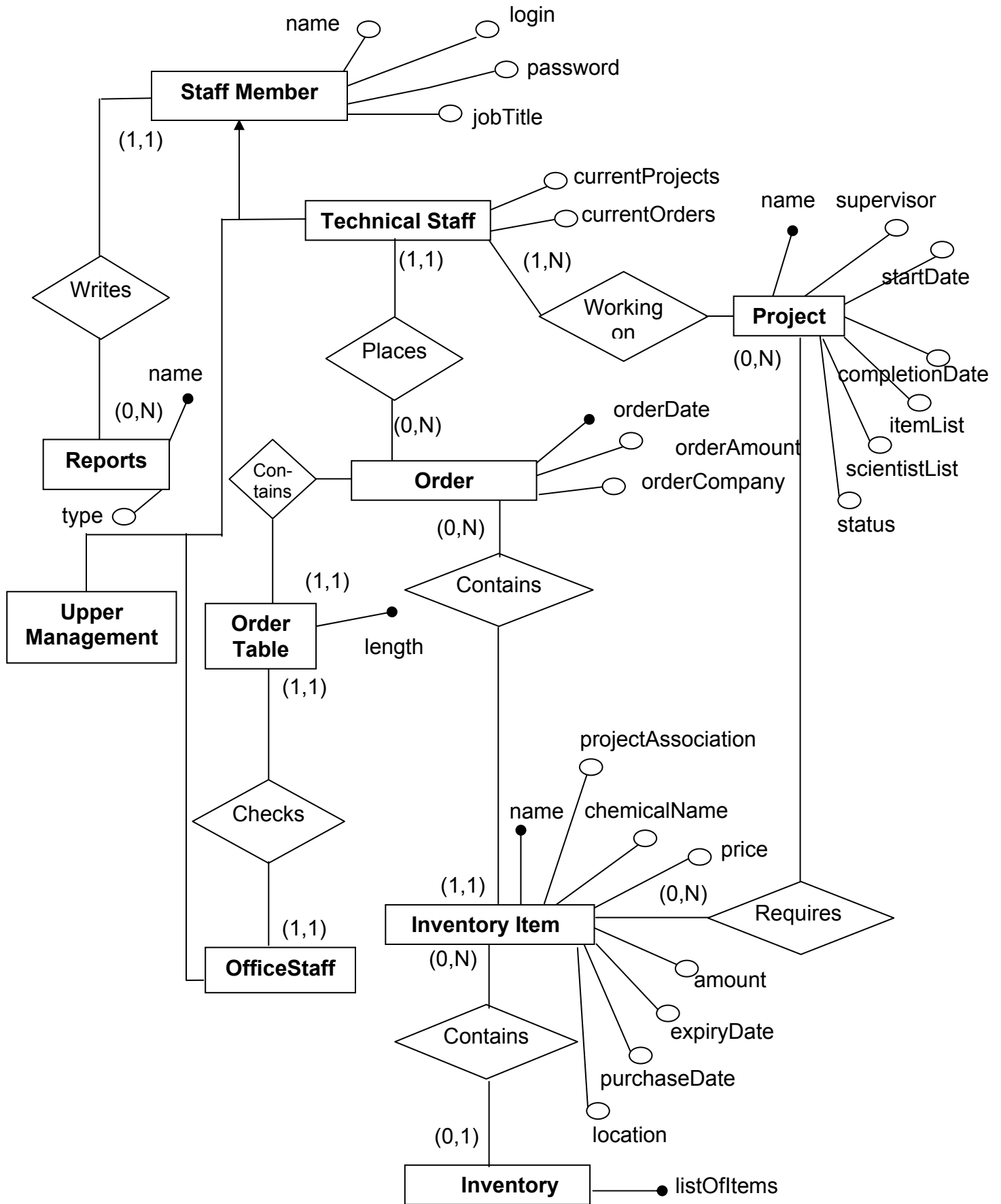| Entity	Description	Attributes	Identifier
Staff Member	Employee of PRO	name, login, password, jobTitle	login, password
Upper Management	Managers, President, etc.	-	login, password
Technical Staff	Scientists working on research projects. Technicians, Technologists, etc.	name, login, password, jobTitle, currentProjects, currentOrders	login, password
Office Staff	Workers handling accounting, administration, etc.	name, login, password, jobTitle	login, password
Order Table	A list of all the orders the technical staff enter into the system.	length	length
Order	A request that technical staff enter into the system for some material.	orderDate, orderAmount, orderCompany	orderDate
Reports	Summaries providing information on the projects the employees are working on, personnel, etc.	name, type	name
Projects	Projects the technical staff work on.	name, supervisor, startDate, completionDate, itemList, scientistList, status.	name
Inventory	A complete list of the current available inventory at the company	listOfItems	listOfItems
Inventory Item	A particular item in the inventory.	name, projectAssociation, chemicalName, price, amount, expiryDate, purchaseDate,location	name

Relationship	Description	Entities Involved	Attributes
Writes	Associates a staff member with a report.	Staff Member(1,1), Report(0,N)	-
Places	Associates technical staff with an order.	Technical Staff(1,1), Order(0,N).	-
Working On	Associates technical staff with a project.	Technical Staff(1,N), Project(0,N).	-
Checks	Associates office staff with order table.	Office Staff(1,1), Order Table(1,1).	-
Contains	Associates order with inventory item.	Order(0,N), Inventory Item(1,1).	-
Contains	Associates inventory with inventory item.	Inventory(0,1), Inventory Item(0,N).	-
Requires	Associates project with inventory item.	Project(0,N),Inventory Item(1,1).	-

Business Rules

Constraints

(BR1) Only technical staff can place an order.
(BR2) Only office staff can satisfy the order by purchasing the required material.
(BR3) Each project must have one or more technical staff (as supervisor) working on it.
(BR4) Only technical staff can work on projects.

Derivations

(BR5) The cost of a project is calculated as the total of multiplying each item in the inventory by its respective price.
(BR6) Each technical staff member's expenditure is calculated as the total of multiplying each item that he/she has ordered by its respective price.
(BR7) The present amount available for each item in the inventory is calculated by adding the amount present in storage and the amount returned by the various technical staff at the end of the experiments.

Appendix 22

Software Architecture – April 10th 2001

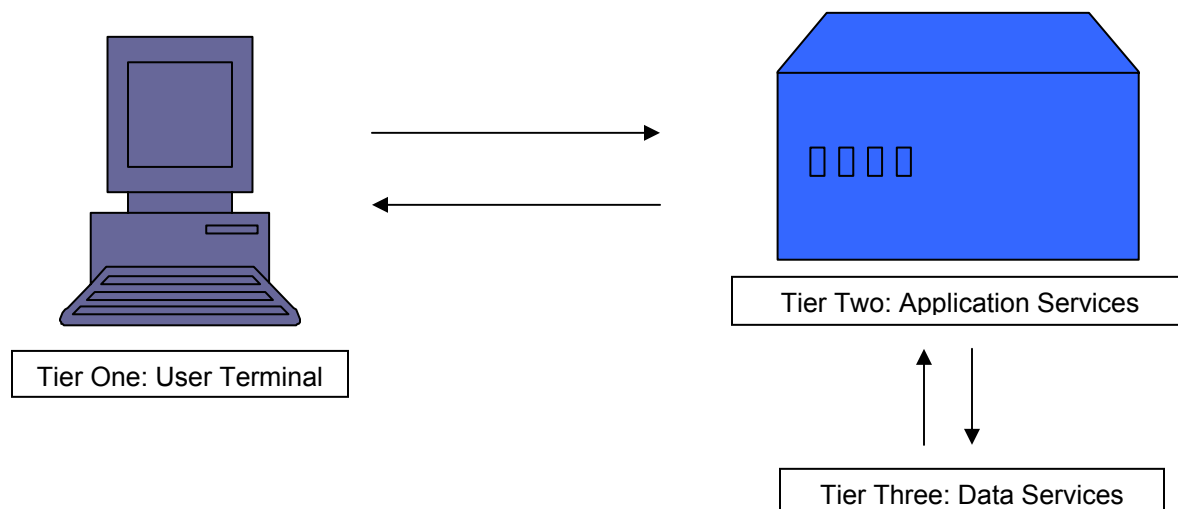Notes: Analysis and selection of a software architecture for our new system.

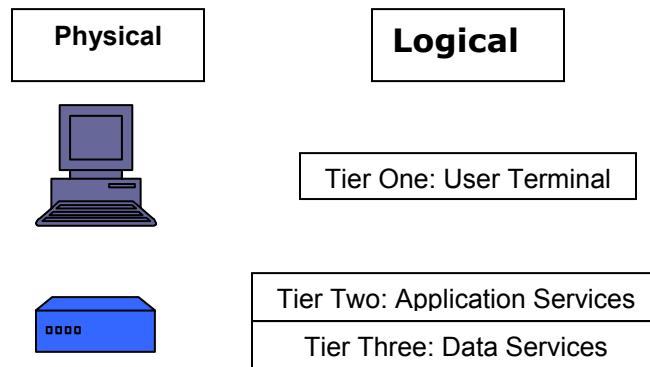Summary: After our analysis, we elected to adopt a repository-based architecture.

Architecture Selection

Although we considered many possible architectures, there were two alternatives that our team debated: the Three-tier Client Server Architecture and the Repository-Based Architecture. They both seemed fairly appropriate, so we present them both here:

Three-Tier Design:
The three-tier model is a type of client server architecture. This model consists of two types of machines, service consumers (clients) and service providers (servers). What is most important about this architecture is that clients and servers may or may not be running on dedicated machines. Thus it is possible for a machine to act as both a client and a server depending on the situation. In the case of PRO, the clients will be the individual microcomputers all around the company and the central server will be the service provider. Information exchange between the machines is done through messages requesting data, or invoking procedures on the remote machine. In this case it will usually involve one of the clients requesting information from the server. The three tiers of the system will separate the user (client) from the complex inner workings of the software. The first tier is an application layer that the user interacts with. Under the application layer is a processing layer that processes the user's queries. Under that is the database layer which receives the processed queries and returns the appropriate data to the processing layer. The processing layer passes the data back to the application layer which formats it for the user to see. The layers will exist on different machines; the processing and database layers will exist on the server, while the application layer will exist on each client machine.

Tier One: User Terminal

Tier Two: Application Services

Tier Three: Data Services

Physical

Logical

Tier One: User Terminal

Tier Two: Application Services

Tier Three: Data Services

Repository-Based Design

Repository based architecture is very similar to the client-server model described above. The system revolves around a central data structure (usually a database), and a collection of independent components which operate on the central data structure. One of the only main differences between this and the three-tier model is that in this case there is a dedicated server and dedicated clients. Under no circumstances can a client machine become a service provider, thus the system is very directed and straightforward. In the case of PRO, the central data structure would be the inventory database, which would be located on the server. Each client would access the server over the network and query the data they require. All software procedures would be present on the server and accessible from any client machine.

Repository

Sub-Application for Office Staff

Sub-Application for Technical Staff

Terminal

Physical

Logical

User Terminal

Database Repository

We decided that a repository-based system would be most appropriate for the software we are trying to design. The central data structure would be the inventory database and there would be multiple dedicated client machines all capable of accessing this repository. The database could be designed and installed directly onto the server and be instantly accessible from any computer on the network. We felt that there was no need to complicate the situation by creating multiple tiers and having any computer capable of being the service provider. This "one-way" access system also provides a level of security (since no one will have access to any other machine on the network except the server). It also greatly simplifies the construction of the system for the developers and the use of the system for PRO employees.

Appendix 23

Software Design: Systems and Subsystems – April 10th 2001

Notes: Division of the Inventory system into subsystems and modules.

```
                        ┌──────────────────┐
                        │ Inventory System │
                        └──────────────────┘
            ┌──────────────────┼──────────────────┐
   ┌────────────────┐   ┌──────────┐    ┌───────────┐
   │ Administration │   │ Ordering │    │ Reporting │
   └────────────────┘   └──────────┘    └───────────┘
    ┌───────┐ ┌───────┐  ┌────────┐   ┌─────────┐┌──────────┐┌─────────┐
    │Personnel││Project│ │Inventory││Order │  │Personnel││Inventory ││Project  │
    └───────┘ └───────┘  │List    ││Table │  └─────────┘└──────────┘└─────────┘
                          └────────┘└──────┘
```
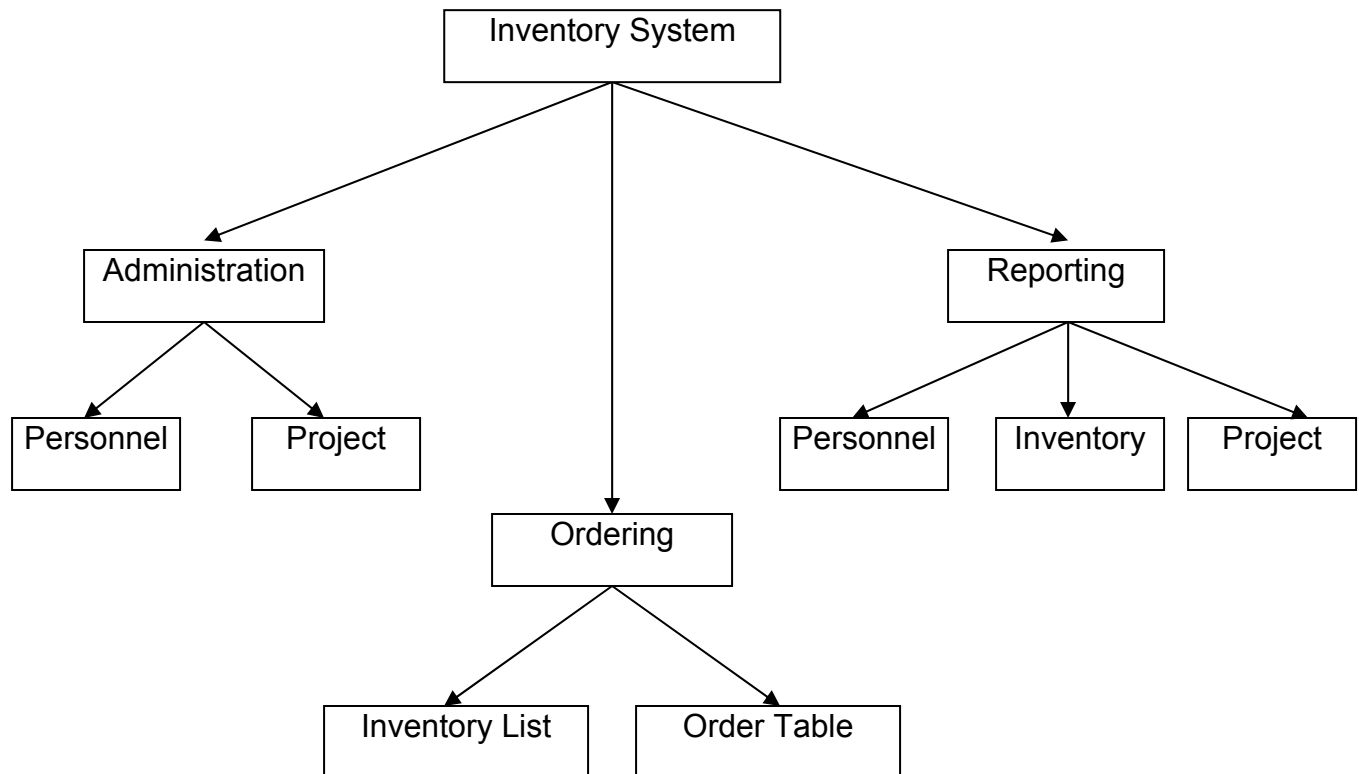
Main System

Inventory System – This encompasses the whole system.

Subsystems

Administration – This is a subsystem designed to handle administrative details mainly dealing with the personnel and project issues.
Ordering – This subsystem will handle all inventory item and ordering functions.
Reporting – This subsystem will deal with the production and printing of all reports in both virtual and hard copy. All report information will be gathered from one of the modules in the system.

Modules

Personnel (Administration) – This module handles all personnel issues in the system. This includes addition of new personnel to the database, correcting the spelling of names, updating personnel projects and the removal of personnel from the system.

Project (Administration) – This module handles all project issues in the system. This includes project names, supervisors, associated staff, inventory lists, start dates, finish dates, current orders, costs, etc.
Inventory List – This module will handle all the inventory information. It will keep track of inventory items, names, amounts, dates, associated projects, associated scientists, etc.

Order Table – This module will administrate the order table. It will accept orders from the Technical Staff, collate and combine orders, create the order table seen by the Office Staff, etc.
Personnel (Reporting) – This module handles the generation of personnel reports. It will gather information from throughout the system and display information such and personnel names, projects, orders, money spent, etc. Most of this information can be gathered from the Personnel (Administration) module. It will also handle the printing of the reports both in virtual and hard copy.
Inventory (Reporting) – This module handles the generation of inventory reports. It will gather information from throughout the system and display information such as item names, amount left, amount ordered, cost, associated projects, associated scientists, etc. Most of this information can be gathered from the Inventory List module. It will also handle the printing both in virtual and hard copy.

Project (Reporting) – This module handles the generation of project reports. It will gather information from throughout the system and display information such as project names, supervisors, associated scientists, items used, total cost, etc. Most of this information can be gathered from the Project (Administration) module. It will also handle the printing both in virtual and hard copy.

Appendix 24

Database Architecture – April 11th 2001

Notes: Analysis of possible database architectures.

Since the Feasibility Study phase, it has been our intention to design a relational database with MS Access. Thus our database will be composed of a series of tables containing the data we require. We will indeed use MS Access to design these tables and relations (for software platform decisions, see Appendix 12). We did look at the hierarchical and network models, but quickly rejected them both. Our data did not fit nicely into the hierarchical format i.e. there was no clear 'parent-child' link between any of our data. The network model was a possibility, but the implementation seemed too complex and the relational model provided a much simpler alternative.

What follows this appendix is an analysis of our current E-R Diagram and the translation of that model into another that can be properly represented by a relational database.

Appendix 25

Database Design Analysis – April 11th 2001

Table of Volumes

Concept	Type	Volume
Staff Member	E	15
Upper Management	E	2
Office Staff	E	3
Technical Staff	E	10
Reports	E	3/Week
Order	E	30/Week
Inventory	E	500
Inventory Item	E	5/person/day
Project	E	7 on average
Writes (Reports)	R	1/Staff Member/Week
Places (Orders)	R	10/Tech Staff/Week
Checks (Order Table)	R	1/day
Contains (Inventory)	R	300
Contains (Order)	R	1/Order
Requires (Inventory Item)	R	50/Project
Working On (Project)	R	2/Projects/Tech Staff

Table of Operations

Operation	Frequency
Operation 1 – Staff Member – addEmployee	3 times / year
Operation 2 – Staff Member – changePassword	15 / employee / month
Operation 3 – Office Staff– readOrderTable	1/ day
Operation 4 – Office Staff– dispatchOrders	1/ day
Operation 5 – Order Table– collateOrders	1 / day
Operation 6 – Order Table– generateTable	1 / day
Operation 7 – Order Table– addToTable	5 / day
Operation 8 – Order Table– checkLength	3 / week
Operation 9 – Reports– printHardReport	3 / week
Operation 10 – Reports– generateReport	3/ week
Operation 11 –Technical Staff– pickUpItem	5 / day / Tech Staff
Operation 12 –Technical Staff– addProject	1 / every 2 months
Operation 13 –Technical Staff– removeScientist	3 / year
Operation 14 –Technical Staff– addScientist	3 / year
Operation 15 –Technical Staff– changeProjects	6 / year
Operation 16 – Order – placeOrder	5 / day / Tech Staff
Operation 17 – Order – cancelOrder	2 / week
Operation 18 –Inventory Item – checkLocationItem	5 / day / Tech Staff
Operation 19 –Inventory Item – checkAvailability	5 / day / Tech Staff
Operation 20 –Inventory Item – updateItem	5 / day / Tech Staff
Operation 21 –Inventory Item – addItem	5 / day / Tech Staff
Operation 22 –Inventory Item – removeItem	5 / day / Tech Staff
Operation 23 –Inventory Item – checkOutItem	5 / day / Tech Staff
Operation 24 –Inventory Item – changeProjectAssociation	1 / every 2 months
Operation 25 – Project – startProject	1 / every 2 months
Operation 26 – Project – completeProject	1 / every 2 months
Operation 27 – Project – updateItemList	3 / day
Operation 28 – Project – updateScientistList	1 / every 2 weeks
Operation 29 – Project – updateStatus	3 / day

Table of Accesses, with Redundancy

Operation 1

Concept	Type	Accesses	Type
Staff Member	Entity	1	W

Operation 2

Concept	Type	Accesses	Type
Staff Member	Entity	1	W

Operation 3

Concept	Type	Accesses	Type
Office Staff	Entity	1	R
Checks	Relationship	1	R
Order Table	Entity	1	R

Operation 4

Concept	Type	Accesses	Type
Office Staff	Entity	1	R
Checks	Relationship	1	R
Order Table	Entity	1	R

Operation 5

Concept	Type	Accesses	Type
Order Table	Entity	1	R
Contains	Relationship	3	R
Order	Entity	3	R

Operation 6

Concept	Type	Accesses	Type
Order Table	Entity	1	R
Contains	Relationship	3	R
Order	Entity	3	R

Operation 7

Concept	Type	Accesses	Type
Order Table	Entity	1	W
Contains	Relationship	1	R
Order	Entity	1	R

Operation 8

Concept	Type	Accesses	Type
Order Table	Entity	1	W

Operation 9

Concept	Type	Accesses	Type
Reports	Entity	1	R

Operation 10

Concept	Type	Accesses	Type
Reports	Entity	1	R

Operation 11

Concept	Type	Accesses	Type
Technical Staff	Entity	1	R
Working On	Relationship	1	W
Project	Entity	1	W

Operation 12

Concept	Type	Accesses	Type
Technical Staff	Entity	1	W
Working On	Relationship	1	W
Project	Entity	1	W

Operation 13

Concept	Type	Accesses	Type
Technical Staff	Entity	1	W
Working On	Relationship	1	W
Project	Entity	1	W

Operation 14

Concept	Type	Accesses	Type
Technical Staff	Entity	1	W
Working On	Relationship	1	W
Project	Entity	1	W

Operation 15

Concept	Type	Accesses	Type
Technical Staff	Entity	1	W
Working On	Relationship	1	W
Project	Entity	1	W

Operation 16

Concept	Type	Accesses	Type
Order	Entity	1	W
Contains	Relationship	1	W
Order Table	Entity	1	W

Operation 17

Concept	Type	Accesses	Type
Order	Entity	1	W
Contains	Relationship	1	W
Order Table	Entity	1	W

Operation 18

Concept	Type	Accesses	Type
Inventory Item	Entity	1	R
Contains	Relationship	1	R
Inventory	Entity	1	R

Operation 18

Concept	Type	Accesses	Type
Inventory Item	Entity	1	R

Operation 19

Concept	Type	Accesses	Type
Inventory Item	Entity	1	R
Contains	Relationship	1	R
Inventory	Entity	1	R

Operation 20

Concept	Type	Accesses	Type
Inventory Item	Entity	1	W
Contains	Relationship	1	W
Inventory	Entity	1	W

Operation 21

Concept	Type	Accesses	Type
Inventory Item	Entity	1	W
Contains	Relationship	1	W
Inventory	Entity	1	W

Operation 22

Concept	Type	Accesses	Type
Inventory Item	Entity	1	W
Contains	Relationship	1	W
Inventory	Entity	1	W

Operation 23

Concept	Type	Accesses	Type
Inventory Item	Entity	1	W
Contains	Relationship	1	W
Inventory	Entity	1	W

Operation 24

Concept	Type	Accesses	Type
Inventory Item	Entity	1	W
Requires	Relationship	1	W
Project	Entity	1	W

Operation 25

Concept	Type	Accesses	Type
Project	Entity	1	W

Operation 26

Concept	Type	Accesses	Type
Project	Entity	1	W

Operation 27

Concept	Type	Accesses	Type
Project	Entity	1	W
Requires	Relationship	1	W
Inventory Item	Entity	1	W

Operation 28

Concept	Type	Accesses	Type
Project	Entity	1	W
Working On	Relationship	1	W
Technical Staff	Entity	1	W

Operation 29

Concept	Type	Accesses	Type
Project	Entity	1	W
Working On	Relationship	1	W
Technical Staff	Entity	1	W

Appendix 26

Schema Translation and the Relational Model – April 13th 2001

Notes: Relational model and modified E-R Diagram for the Inventory System.

In translating the database schema we derived previously (see Appendix 25), there were many factors to consider. First we had a few redundancies to deal with in our initial design. For example, projects kept a list of inventory items they used, and inventory items kept track of their project associations. Some redundancies have been removed now, but a few still remain. In many cases, we felt that these redundancies were best left in place i.e. their presence will make the system work faster (quicker data search and retrieval times). The storage space they consume is negligible considering that we have reserved 300MB of disk space for this database (much more than a database of this magnitude requires). The generalization relationship between the staff members was also eliminated. Since different types of staff members had different operations, it was impossible to combine then into a single entity. Therefore they were left as separate entities and each one is pictured with its own set of staff member attributes. No partitions or mergers were performed on any of the entities. It was felt that each entity represented much too distinct a concept for there to be mergers. On the other hand, each entity was very concisely described, so no partitions were thought necessary.

Relational Model:

Technical Staff (login, password, jobTitle, currentProjects, currentOrders)

Order(orderDate, technicalStaff, inventoryItem, orderAmount, orderCompany)

Reports(name, type, author)

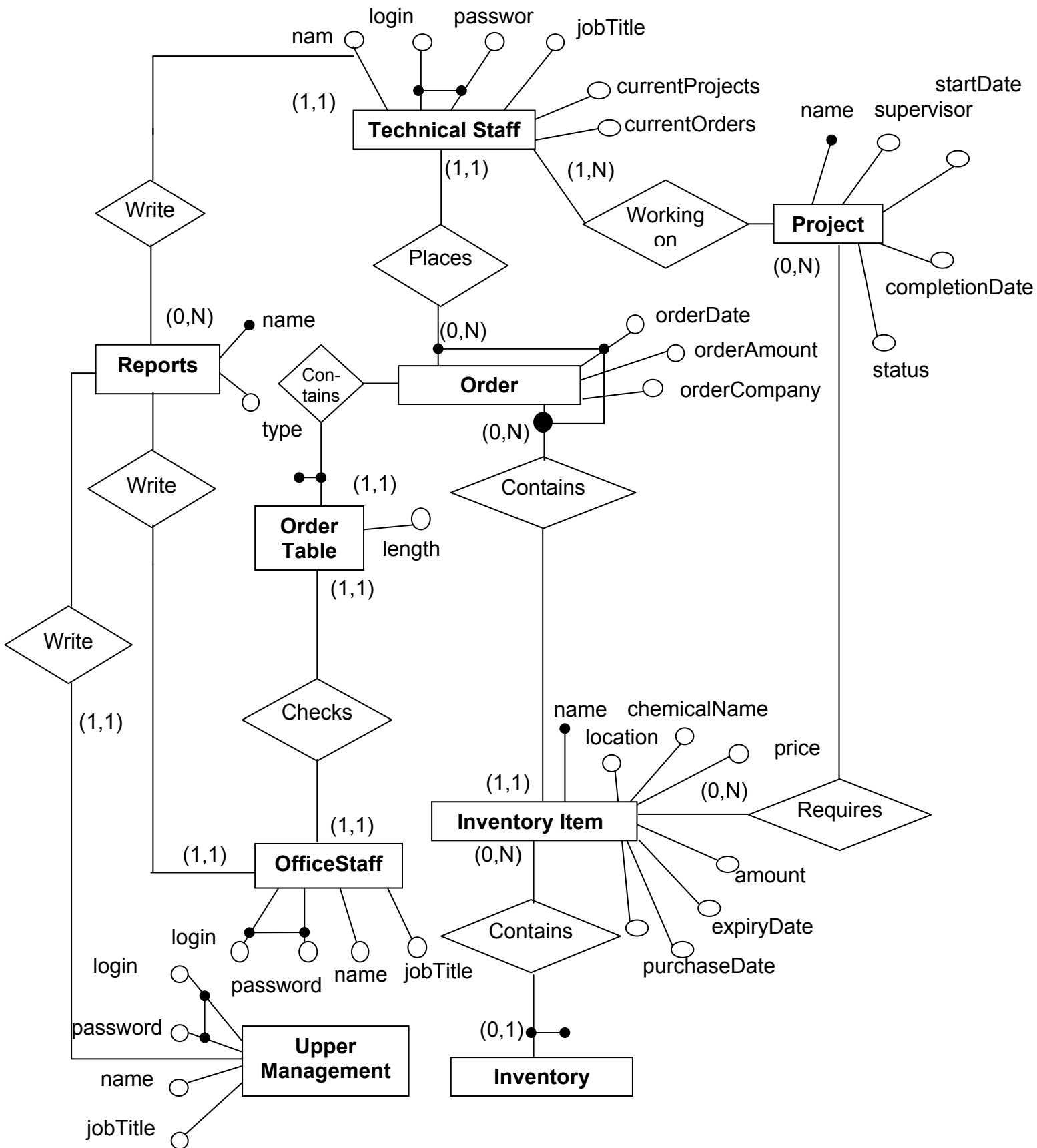Project(name, supervisor, startDate, completionDate, status, inventoryItem, technicalStaff)

Order Table(order, length)

Inventory Item(name, chemicalName, price, amount, expiryDate, purchaseDate, location)

Office Staff(login, password, name, jobTitle)

Inventory(inventoryItem)

UpperManagement(login, password, name, jobTitle)

Appendix 27

User Interface Design – April 13th 2001

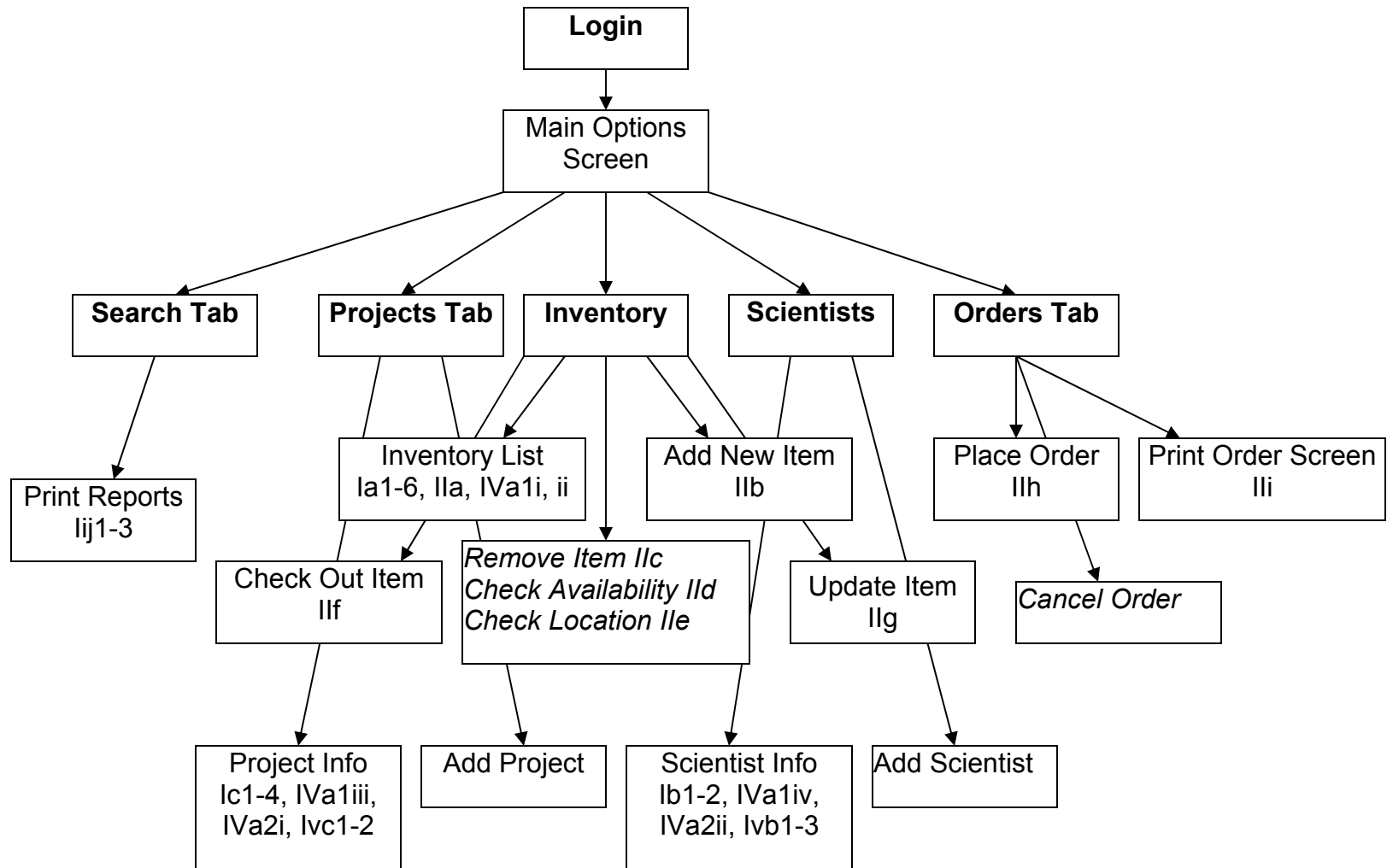Notes: Discussion of user interfaces with regards to listed requirements

The user interface design was very thoroughly considered since the success of this system relies very heavily on user acceptance. As mentioned previously, most users of this system will be novices and thus a simplistic and intuitive interface is mandatory. As such, we decided that each screen should present the user with as little information and as few choices as possible. In this manner, the users will not be inundated with data; they will simply be presented with a small number of choices that they can select from in order to get the information they need. This fulfills the non-functional user interface requirements as laid out in section V (see Appendix 6).

The first screen the users will be presented with is a login and password screen. All employees are issued logins and passwords so they can access the network from the computers in their lab or office. Users can use this information to log onto the system and then change their passwords once they are inside. This takes care of the security requirements laid out in section III of the non-functional requirements (see Appendix 6). This screen is actually illustrated in Appendix 28.

Once they have successfully entered the system, users will be presented with a screen composed of multiple "tabs" which look similar to the tabs where labels are put on paper files. These tabs bear various labels such as "projects", "orders", etc.; clicking on any of them brings up a different screen with information appropriate to that subject. For example, the "orders" screen will have information about all the current orders in the system along with buttons that will spawn new screens allowing users to enter new orders, cancel orders, etc.

Our list of functional requirements specifies a list of data we are required to be able to accept from the user, or display to the user. The following is a tree of screens and functions that our interface will incorporate in order to facilitate this input/output (I/O) of information. Items appearing in boxes, written in regular print are screens that a user can navigate to i.e. there will be a 'Login' screen, and a 'Main Options' screen. Items written in italics will not appear on their own screen, rather they will be accessible through buttons present on the screen that points to them i.e. the 'remove item' function will be accessible through a button on the 'inventory' screen. The numbers after the screen or function indicate the functional requirement the information they contain will fulfill (see Appendix 5). The connections between boxes illustrate the path a user would have to take through the system in order to obtain that data or execute that function. Items that are in bold are illustrated in Appendix 28. Of course what follows is not a

comprehensive list of all possible interactions with the system, but it provides an overview of the major functions that will be available.
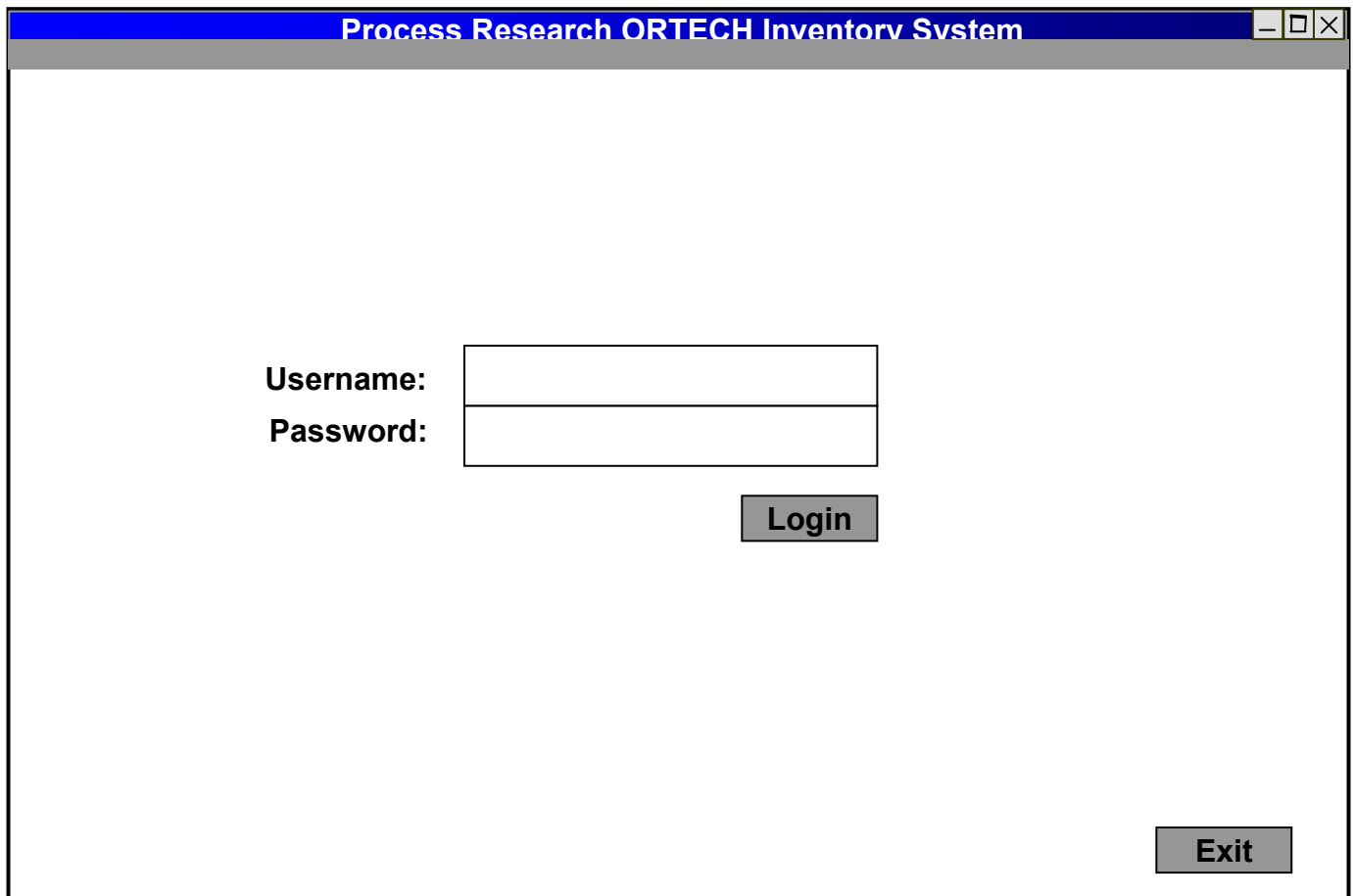
```
                              ┌──────────┐
                              │  Login   │
                              └────┬─────┘
                                   │
                          ┌────────▼────────┐
                          │  Main Options   │
                          │     Screen      │
                          └────────┬────────┘
```

Login

Main Options Screen

Search Tab

Projects Tab   **Inventory**   **Scientists**   **Orders Tab**

Print Reports
Iij1-3

Inventory List
Ia1-6, IIa, IVa1i, ii

Add New Item
IIb

Place Order
IIh

Print Order Screen
IIi

Check Out Item
IIf

Remove Item IIc
Check Availability IId
Check Location IIe

Update Item
IIg

Cancel Order

Project Info
Ic1-4, IVa1iii,
IVa2i, Ivc1-2

Add Project

Scientist Info
Ib1-2, IVa1iv,
IVa2ii, Ivb1-3

Add Scientist

77

Appendix 28

Screen Designs – April 11th 2001

Notes: Rough samples of potential user interface screens. This does not constitute a complete set of screens.

Login –
This is a simple login screen that logs the user in as a Technical Staff, Office Staff or Upper Management member, depending on their username and password. The authentication is carried out by Microsoft Windows 2000's authentication server. All employees have usernames and password to use the network already.

Process Research ORTECH Inventory System	_ □ ✕

Username:

Password:

Login

Exit

Searching

Notice the interface is composed of "tabs" running along the top of the screen. Depending on which group the user belongs to, the particular tabs will be enabled or disabled, for example a Technician cannot access the projects tab to add and remove projects or personnel from the database (only project supervisors can). This is a sample screen that an office staffer might be faced with. The office staff can view inventory usage based on projects, personnel etc. An office staff member can enter a chemical name and the project it's being used in, this will query the database to find out how much of that chemical was used in that project. At least one of the fields has to be filled in or the search cannot be completed. If the office staff member just wants to see the entire database without entering a search query, the "Show complete database" function can be used. The Print reports button will create reports based on the information entered. It will bring up another dialog box that allows selection of the format of the report and other details.

Process Research ORTECH Inventory System	_ □ ×

File Edit View Insert Tools Window

Search Database	Projects	Inventory	Scientists	Orders

Specify   Search

Chemical	Sulphuric Acid
Formula	
Project	HGENG001A
Date	
Personnel	
Cost	

Show complete   **Search**   **Clear**   **Print Reports**

You are logged on as *username* of group OfficeStaff

Logoff   **Exit**

Project Administration

The projects tab facilitates the addition and removal of projects. It also has a list of inventory items and the various details of the project. Users will be able to sort the list of projects by clicking on the column heading i.e. click on 'name' to sort projects alphabetically. Notice the bottom left side of this screen. A similar bar appears on every screen except the login screen. Only certain users can access this page; logins identify user types to the system and prevent access to unauthorized pages.

Process Research ORTECH Inventory System						_ □ ×
File Edit View Insert Tools Window						

Search Database	Projects	Inventory	Scientists	Orders

Project Name [   ]   **Add/Remove Project**

Name	Supervisor	Start Date	Completion Date	Unfilled Orders	Items Currently Used	All items used

You are logged on as _username_ of group OfficeStaff   **Logoff**   **Exit**

80

Inventory Administration

The Inventory tab, is available to both technical and office staff. With this screen they can add and remove inventory items, check out and update the amounts of items used and remaining. Pressing the buttons will bring up other dialog boxes that confirm or get other needed information to complete the task. Again, users will be able to sort the table by clicking on column headings.

							_ □ ✕
Process Research ORTECH Inventory System							

File   Edit   View   Insert   Tools   Window

Search Database	Projects	Inventory	Scientists	Orders

Item Name   Sulphuric Acid   **Add/Remove**   **Update**

Check Out

Name	Chemical Names	Amount	Current Projects	Expiry Date	Date entered into inventory	Date of use	Cost per unit measure

You   are   logged   on   as   _username_   of   group   **Logoff**   **Exit**

Employee Administration

The Scientists tab is used to add scientists to the database as well as assign and un-assign scientists from various projects. This screen is useful in keeping track of what scientists are working on, how many orders they have placed, how much money they are spending, etc.

Process Research ORTECH Inventory System			_ □ ✕
File Edit View Insert Tools Window			

Search Database	Projects	Inventory	Scientists	Orders

Scientist   Marion Gotts   **Add/Remove**

Proiect   HGENG001A   **Assign / Unassign**

Name	Current Project	Unfilled Orders

You   are   logged   on   as   _username_   of   group   **Logoff**   **Exit**

Ordering

The Orders tab is used to place orders or view the order table. Technical staff will see only orders they have placed on the table while the office staff will see all orders placed since the last dispatching. Thus technical staff will only be able to remove their own orders, whereas office staff will be able to remove any order (i.e. when they are dispatching it). There is an option to print the table out as a hard copy as well. Clicking the place order button will pop up another screen where the particulars can be entered.

Process Research ORTECH Inventory System `_ □ ×`

File   Edit   View   Insert   Tools   Window

| Search Database | Projects | Inventory | Scientists | Orders |

Item Name   `Sulphuric Acid`   **Place Order**   **Cancel Order**

Item Name	Ordering Personnel	Associated Project	Amount	Date	Cost

Print Table   **Remove**

You   are   logged   on   as   *username*   of   group   **Logoff**   **Exit**

Appendix 29

Work Division

Andrew: Interviewing, concepts and planning, State/ Activity Diagrams
Manojav: Interviewing, formatting, and typesetting, Screen Displays
Kunendran: Diagrams, Case Diagrams and other text
Junaid: E-R Diagrams, Database Schema, Database Operations.

Name	Student Number	% of Effort	Signature
Andrew Ramadeen	971263350	25.0	
Kunendran Deivendran	971273750	25.0	
Junaid Yousuf	98	25.0	
Manojav Sridhar	971263500	25.0	