CSC 340
Assignment Two
Laboratory Animal Services Of Cornell University

Team 3
David Gilbert
Anna Krolikowska
February 1st, 2002
Professor: John Mylopoulos
TA: Attila Barta

Table of Contents

Organization 3
The Current System 3
A Problem 4
Preliminary Solution Requirements 4
Proposed Alternatives 5
Detailed Description of the Selected Alternative 6
System Requirements 8
Functional Requirements 8
    LAS Remote Location Subsystem 10
    Accounting Office Subsystem 10
    LAS Management Subsystem 11
    Printer Subsystem 11
    Software Maintenance Subsystem 12
    Data Base Administration Subsystem 12
    Client Subsystem 13
Non-Functional Requirements 14
    Interface Requirements 14
    Performance Requirements 15
    Operating Requirements 16
    Lifecycle Requirements 17
    Economic Requirements 17
    Human Factors 17
    Testability 17

Appendices:

Appendix A-Use Cases 19
    Lab Technician 19
    Accountant 20
    Las Manager 21
    Client 22
    Printer 22
    Software Manager 23
    Database Administrator 23
    Bigger Picture 24
    Use Case Descriptions 25
Appendix B-Class Diagrams 27
    Data Dictionary 28
Appendix C-Sequence Diagram 36
    Providing Services to Client 36
    Census Data Entry 36
    Bill Information Extraction 37

Bill Processing 38
Appendix D-State Diagrams 39
Accounting Office 39
Remote Location 40
Researcher 41
Appendix E-Correspondence and Contacts 42

ORGANIZATION

Laboratory Animal Services (or LAS) is a unit within the College of Veterinary Medicine, at Cornell University, that provides specialist services to faculty members who use animals for teaching or research. Essentially it provides physical and human resources for the excellent care and health maintenance of animals used in teaching and research at the College of Veterinary Medicine.

The unit maintains several different animal facilities. Each facility is dedicated to a species of animal and varies in terms of sophistication of facilities and resources. For example, horses may be kept on pasture with run-in sheds, in stalls, or in high security stalls for work with infectious diseases. LAS also has facilities for veterinary care of these animals and even surgical facilities. LAS provides housing, care, and veterinary services to several species including mice, rats, guinea pigs, rabbits, cats, dogs, sheep, cattle, horses, pigs, fish, birds (including poultry) and more.

The nature of the services ranges from basic food and care, routine observation for general health, consultations with veterinarians specializing in laboratory animal care, etc. In some cases, LAS provides all of the veterinary care of animals. (In others, where the researcher or teacher is a veterinarian, that person may elect to provide veterinary care him or her self.) The Unit finds itself responsible for advising investigators and teachers on the relevant local, state and federal regulations, and also responsible for making sure that those regulations are not breached. (This is sometimes quite a difficult position for them.) So, they will help with writing of animal care and use protocols, and then scrutinize those to make sure they comply with regulations and are passed by the Institutional Animal Care and Use Committee (a federally-mandated committee).

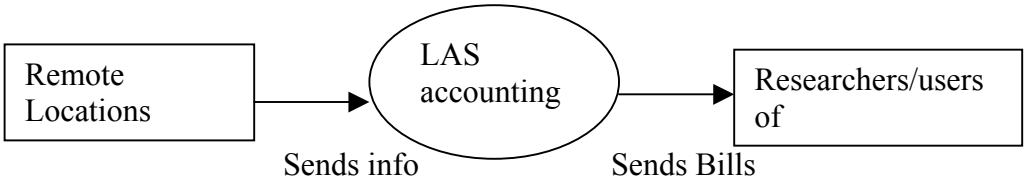
Once the paper work is in order, LAS will order or procure animals from vendors, arrange for transport, acclimate animals to new surroundings, house feed and care for the animals. They also provide some technical support. For example, relatively few research technicians know how to hold and handle mice or rats so that neither they nor the animals get hurt, etc. After the research or teaching is over, LAS will help with adoption, sale, etc, of the animals.

Financially, LAS is a break-even operation. The law does not allow them allowed to make or lose money. This makes accounting and efficiency very important.

THE CURRENT SYSTEM

Researchers that use LAS facilities are billed on a monthly basis for the services provided. Census data from each facility is compiled manually onto handwritten forms that are then sent to the centralized accounting office. The handwritten forms are then entered into an Access database containing rates by species, researcher name, and general ledger charging data. When the keying in of this information is complete, the data is exported to an Excel spreadsheet to verify that the rates charged by species are correct. Two copies of the invoices are then printed. One is mailed to the accounting office of the researcher that is being billed, and the other is attached with the corresponding handwritten forms, and filed away. Customers have one week to respond with questions, changes in general ledger attributes, etc. The bills are done by animal type, and housing

type (some are special cases, e.g. isolation). Daily rates are established in advance for each species.



A PROBLEM

The fact that all of the census and billing information is originally done manually, instead of digitally, creates some problems. These handwritten notes are often hard to read. Once the notes reach the main accounting office all the data is keyed into a computer. An accounting staff who has no idea how the LAS facilities actually work does this keying. This is obviously very problematic. Due to the fact that the staff does not understand the work and process of the facilities, hard to read and/or ambiguous notes can lead to incorrect data being entered into the system. Also, often times the billing forms that LAS staff fill out at all of the facilities are not set up to capture enough data in some cases. This leads to more confusion and ambiguity.

It is clearly very time consuming for all of the data from all of the separate facilities to be transported and keyed into the computer by one person. It is not very time or cost efficient. This is especially a problem when money is so tight and the budget has to be perfectly balanced (due to the fact that it is a non-profit unit). The financial position of the unit needs to be improved, and an optimization of this billing system would contribute substantially.

PRELIMINARY SOLUTION REQUIREMENTS

Clearly the system needs to be improved. The solution requirements, as outlined immediately following, are merely a preliminary sketch of what is needed based on e-mail correspondence with Cornell Accountant Timothy Pollard (see **Appendix E**).

The nature of the problem calls for a computerized system that will eliminate the handwritten notes and the confusion and cost that directly result. According to Mr. Pollard (see **Appendix E.4**), there are several requirements for a new computerized system. First, all information should be stored in some sort of centralized manner so that information can be added and accessed from multiple locations (the multiple LAS facilities). Secondly this information needs to be secure. Only the specified facilities associated with LAS should be able to submit and withdraw information from the system, no other outside access to the system should be possible. There should also be the

capacity for multiple concurrent users. Staff at any of the facilities should be able to access the information at any time, and possibly therefore, at the same time. Lastly there should be some sort of query capabilities to find information stored on the database quickly and easily. The possibility should exist to search for information based on several key fields including for instance researcher name, animal type, type of housing facilities, etc.

An adopted solution should ultimately eliminate all of the confusion and inefficiency associated with handwritten notes being written and interpreted at multiple stages and locations of the overall process. Ideally the necessary information is entered in once, on site at the facility. From that initial data, the production of all other necessary forms (such as bills, inventories, census, etc.) should be possible. All changes to the initial data can just be made on the system as opposed to having to re-enter a lot of information again.

For the management this type of solution would save a lot of time and money. You would not need to take so much time in simple data input and manipulation, and as a result would not have to pay people do it. Employees could spend their time doing more important and productive things.

For the researchers that use LAS this would also be a great improvement. Currently if a researcher has animals at multiple facilities he would receive multiple bills since independent, handwritten, invoices are prepared at each location and processed separately at the centralized accounting office. With the outline to a solution as described above, each facility, when inputting data, could enter it straight into the appropriate researchers file, creating a single bill rather than several that the researchers have to deal with and pay separately.

PROPOSED ALTERNATIVES

In our Feasibility Study we have proposed 4 alternatives, which mostly fulfill the general requirements stated by the LAS unit. The following is the list of the alternatives:

1. ALTERNATIVE #1: WEB-BASED PROCESSING SYSTEM

   Since the data will be sent from different remote locations we can use an Internet-based solution. In this system each lab technician will be required to fill in the online form at the end of his/her shift. Upon sending the form the data will be automatically updated in the local database (on SQL Server) so the LAS accounting unit will have an easy access to this information. This system requires the creation of a local database server at the LAS accounting unit and the development of a web-based software package, which would upload and extract the information present in the database.

2. ALTERNATIVE #2: FTP FILE UPLOADING SYSTEM

   For this solution building a new server system is not required and the LAS unit will use the already available resources and softwares. In this solution each lab

technician will have its own PC with Microsoft Excel installed on it. Each lab will use the same format of a spreadsheet form. The data will be typed into this form by each lab technician. At the end of the night the technician of the last shift will be responsible for sending the Excel file through FTP to the LAS accounting unit. Each morning the LAS accounting unit will collect all the files and export their contents to the Microsoft Access main database.

3. ALTERNATIVE #3: PURCHASE OF COMMERCIAL SOFTWARE

LAS was currently considering a purchase of an existing commercial software specializing in maintaining and directing a typical animal lab facility. From the list of existing software we have chosen two, which are capable of providing a desired solution to the LAS accounting unit:

- GRANITE from Topaz™ Technologies, Inc.
  It is an integrated client/server suite of software products that run on any major database and which can run via Internet connection with no local installation. GRANITE is a flexible and easily customized system that integrates facility management, accounting, cost accounting, and tracking of animal and training records. It allows online animal ordering and census, as well as protocol approval.

- SIRIUS from NTM Consulting Services, Inc.
  It is a high-end integrated database management system, which tracks the purchase and receipt of animals, vendor payments, customer invoicing, and standing and planned orders. The system handles multiple accounting functions by tracking requests for billable and non-billable services and supplies, handling customer invoicing, and cost accounting. The cost accounting of SIRIUS includes determining allocation based on the NIH rate setting manual, payroll tracking, and generating profit/cost statements.

DETAIL DESCRIPTION OF THE SELECTED ALTERNATIVE

The system analysis done the in the Feasibility Study project has helped our team to choose the favoured solution to the system based on the 4 proposed alternatives. This solution is Alternative #1 – WEB-BASED PROCESSING SYSTEM.

To implement this solution the LAS accounting unit will have to create a database server system with SQL DBMS installed on it as well as will have to develop a new web-based software, which will allow the users to access the database form the Internet.

The SQL Server database will contain 2 types of information. The first type will be the clients' information, including their names, addresses, and billing information. The second type will be the information about the services provided to the clients upon which the bills will be generated; this will include the types of services LAS labs provide, the list of research animals used, and the pricing rates used for each service and each species of animals.

Both LAS lab technicians and LAS accountants will have an access to the SQL Server database. Lab Technicians' work will be to enter the animal census data, which basically will consist of the following: the name of clients (researchers and professors) that have used the LAS service, what type of service was provided to them, and what type and what quantities of research species were involved. The accountants' work will be to extract the billing information on monthly basis for each client based on above information.

Our system will require the developing of a web-based software package. The purpose of this package will be to allow the LAS members to access the database from remote locations. There will be 3 major types of users: lab technicians, LAS accountants, and LAS managers. Their reasons for accessing the database will be different. Lab technicians will access the database to type in a daily census; accountants will access the database to extract the clients info and billing info; LAS managers will access the database to extract the clients info and billing info as well as assign new services, species types and pricing rates. So basically each type of user will have different access priorities and database query functionalities assigned to them. The web-based software has to take this into consideration.

Additionally, the software will calculate and determine the billing information automatically and by that will limit the accountants' work to only verifying this information. The software has to have some calculation procedures implemented, which will automatically create and display bills in a formatted style, including all the billing details. These formatted bills can be automatically printed out without further re-typing or re-formatting from the accountant's site.

Another calculation feature that the new software should include is the estimation of the pricing rates, so the LAS monetary benefits obtained form the services provided and the LAS average cost expenses will add up to a net of zero dollars. This is extremely important to LAS since it is a non-profit organization that aims to achieve break-even operations.

Now, using the above description of this alternative and further suggestions from our contact person at LAS we can work out and determine the detailed requirements for the LAS system.

SYSTEM REQUIREMENTS

In the previous section we have provided the general description of the new system and described how it will improve the performance of the LAS billing and information storing functionality. This section has a purpose to describe the actual requirements set on the system. These requirements describe in detail how the system will operate, who will be involved, how the data will be handled, and how the system will interact with the outside world components. These requirements were gathered after numerous electronic communications with Mr. Timothy Pollard (the Cornell general accountant), our contact person at LAS. (once again please refer to **Appendix E**.)

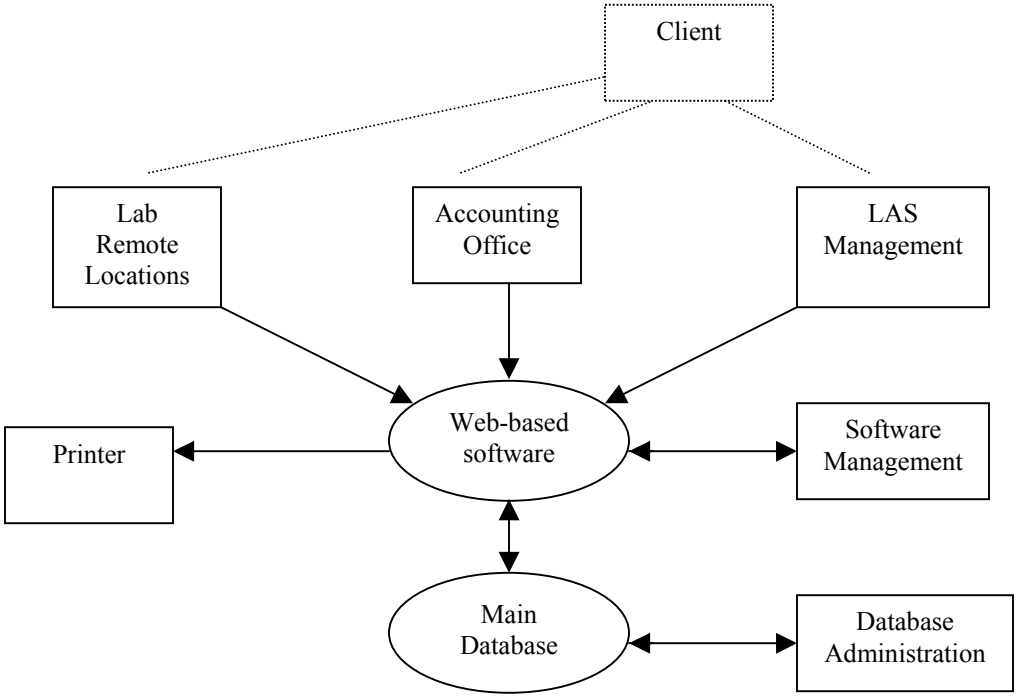
Generally, system requirements are divided into 2 categories: Functional and Non-Functional Requirements. Functional Requirements describe the functions that the new system must support including the activities carried out by the system, the input/ output formats, and how the information is stored and maintained in the system. The Non-functional Requirements describe the quality requirements, which impose global constrains on the system. Our system requirements documentation will include the detail description for both categories of the requirements.

Before we'll get more into details of the system's functionality, let's point out again the general requirements that the system has to meet. These system requirements were brought to our attention in the very first interview with Mr. Timothy Pollard and were since strongly underlined in every telephone or e-mail conversation.
The system should have:

1. Ability of accessing and extracting the information form remote locations.
2. Information has to be stored in a centralized-manner (database).
3. Information accessing has to be secure (password protection).
4. Capability of multiple concurrent users onto the database.
5. Each user should have a different data access priority.
6. Data has to be accessed easily and quickly.
7. Ability of printing formatted bills straight from the system.

FUNCTIONAL REQUIREMENTS

As it has been mentioned above the functional requirements of the LAS system will determine the activities carried out by the system, the information maintained by the system, and the interfaces supported by the system. The diagram shown below has a purpose to provide the general layout of the system's organization, displaying its main components and the connections between them.



As it can be seen from the diagram the "heart" of the system is the Main Database component with its direct connection to the Web-Based software component. The sub-systems which depend on and are connected to the main component of the system (database) are as follows:

- **Lab Remote Locations subsystem**– represents the communication between the animal labs and the main database
- **Accounting Office subsystem** – represents the communication between the accounting office and the main database
- **LAS Management subsystem** – represents the management's control over the system
- **Printer subsystem** – represents the communication between the web-based software and the printer
- **Software Management subsystem** – represents how the web-based application is maintained and managed
- **Database Administration subsystem** – represents how the database is administrated and maintained

Additionally, there is a **Client subsystem**, which is independent of the main database and web-based software. Client subsystem interacts directly with Lab Remote Locations, Accounting Office and LAS Management subsystems.

Note that the web-based software will provide 3 different interfaces based on the application access privileges:

- **Services Information Table Interface** – this interface allows an access to the spread-sheet like information containing the columns representing each client, and

rows representing species types, services provided, and amount of services provided. Each day the lab technicians will enter the census data into this spreadsheet table and upon clicking on the specific button this information will be sent to the database.

- **Billing Information Interface** – this interface will contain the detail billing information for each client. This information is represented in the formatted style, so accountants can read and print bills easily.
- **All Access Interface** – the staff member who will have a privilege to use this interface can access basically every functionality provided by the software including Service Information Table Interface, Billing Information Interface, and Assigning New Clients and Assigning New Pricing Rates Interfaces.

For each of the 7 subsystems presented above we will describe the system requirements concentrating on 3 aspects: system activities, information maintained, and interfaces used by the system.

Lab Remote Locations subsystem:

1) System activities:
    - Lab Technicians provide the animal storing services to the clients
    - Lab Technicians will need to enter their password before accessing database
    - Lab Technicians will enter the data census on daily basis onto the database system through the online software
    - Lab Technicians will access the database for correction and obtaining clients' information

2) Information Maintained:
   The input information is the census data entered by the lab technician. This data represents the services provided by the lab technician to the specific client. This will include the number of services provided, the length of the services, the type and quantity of animals stored in the lab. No output information is returned from the database.

3) Interfaces Used:
   Services Information Table Interface

Accounting Office subsystem:

1) System activities:
    - LAS accountants will need to enter their password before accessing database
    - Accountants have to extract billing information for each client on monthly basis based on the database information
    - Accountants will have to print the bills

- Accountants have to send the bills to the clients
- Accountants allow one week for the clients to respond in case bills are incorrect
- In case the clients respond to the bills, LAS Manger is notified

2) Information Maintained:
   The accountants do not have to enter any data. Their output data will be based on the billing information extracted from the web-based software. This data is compiled from the data previously entered by the lab technicians. The billing data can be stored in the database and/or printed out.

3) Interfaces Used:
   Billing Information Interface

LAS Management subsystem:

1) System activities:
   - LAS Manager will need to enter his/her password before accessing database
   - LAS Manger obtains the general information about the status of the animal services provided and bills sent
   - Based on the information obtained from the database LAS Manager comes up with a new daily rate for each type of animal and service
   - LAS Manager is notified by accountants in case the client responds to the bills
   - In case client responds to the bills, LAS Managers talks to the client directly and corrects the bills if necessary
   - LAS Manager can add, remove, or modify the types of services provided
   - LAS Manager can add, remove, or modify the information about the clients
   - LAS Manger will compare his pricing rates with the rates suggested by the software (the calculated rates aim to achieve the break-even budget for the LAS organization).

2) Information Maintained:
   The input data consists of the billing information, services status, and pricing rates extracted from the database. In this subsystem such information as pricing rates, new services, and new clients can be added to the database. The LAS Manger can print out the extracted information as well.

3) Interfaces Used:
   All Access Interface

Printer subsystem:

1) System activities:

- Printer will wait for the printing jobs which will be assigned by the accountants or LAS manager
- Printer will obtain the printing jobs and extract the printing information from the web-based software

2) Information Maintained:
   The input data consists of the printing jobs assigned by the accountants and LAS managers. The output would be actual bills printed out.

3) Interfaces Used:
   Billing Information Interface

Software Management subsystem:

1) System activities:
   - Software Manager will need to enter his/her password before accessing database
   - Software Manager will observe the behaviour of the software on daily basis
   - Software Managers will be informed about software problems from other staff members and/or from software Error Log file
   - Software Manager will fix the software if any problem is encountered

2) Information Maintained:
   Software Manger will obtain a list of problems from the LAS staff members and/or from the software Error Log file. The status of fixed/unfixed problems will be sent to the LAS Manager.

3) Interfaces Used:
   All Access Interface

Database Administration subsystem:

1) System activities:
   - Database Administrator will need to enter his/her password before accessing database
   - Database Administrator will observe the behaviour of the database on daily basis
   - Database Administrator will be informed about database problems from other staff members and/or form database Error Log
   - Database Administrator will fix the software if any problem is encountered

2) Information Maintained:

Database Administrator will obtain a list of problems from the LAS staff members and/or form database Log Error file. The status of fixed/unfixed problems will be sent to the LAS Manager.

3) Interfaces Used:
   All Access Interface

Client subsystem:

1) System activities:
   - Client will obtain the services from the animal lab
   - Client will be billed for the services on monthly basis
   - If the billing information is incorrect or questions arise, the client will notify the LAS accounting office
   - If the LAS billing office will be informed by the client of the incorrect billing information, the LAS manager will contact the client directly
   - Client can sign up or withdraw form the LAS services at any time

2) Information Maintained:
   Input data will consist of the bills sent from the LAS central accounting unit. The client can accept the bills or enquire/complain about the bills received to the LAS unit. If second scenario applies and the LAS manager agrees there should be corrections to the bill a new invoice will be sent to the client.

3) Interfaces Used:
   This system does not use any of the software interfaces. The communication with LAS is done through the phone, e-mail, or mail.

In most cases it is much easier for the reader to understand the actual activities and connections within the system through visualizing them. For this purpose we have adopted in this document the UML notation, which uses diagrams to represent the system. These diagrams imitate the activities, subsystems, properties of the subsystems, properties of the objects involved, object dependencies, and connections between the objects and subsystems.

The following is the list of diagrams that we will use to demonstrate the system requirements described above:

- **Use Case Diagrams –** These diagrams have a purpose to identify the high level functions within the system (called Use Cases) and to provide a description of the interactions between the users of the system (called Actors). **Appendix A** demonstrates the Use Cases that were identified in our system.

- **Class Diagrams –** These diagrams allow to represent the connections and dependencies between classes (groups of objects with similar properties) within the system. The Class Diagrams for our system are presented in **Appendix B.**

- **Sequence Diagrams –** They have a purpose to show the interactions (Use Cases) exchanged between the system and external objects arranged in a time sequence. **Appendix C** demonstrates the Sequence Diagrams identified in our system.

- **Interaction and State Diagrams –** The purpose of these diagrams is to demonstrate the state transitions of objects within the system. They can be used to describe a lifetime of the objects involved in the system, or interactions of the system within some operating environment. Use **Appendix D** for the system State Diagrams.

Nonfunctional Requirements:

Functional requirements describe the behavior of a system and how the system will interact with the user. In addition to these requirements, one must also take into account constraints and qualities of the system. Constraints are limitations on the system that cannot be negotiated. Time, funding, and the environment the system will operate in all create certain complications that place limitations on various aspects of the system which cannot be compromised on. It is very important that these things be investigated thoroughly prior to the initial design so as to save much time, effort, and resources later on. The qualities of a system are aspects of the system that stakeholders will care about and affect their level of satisfaction with the system. These qualities and constraints are referred to as non-functional requirements, and are outlined below. Much of this information was received from Dr. Gilbert and Timothy Pollard in e-mail and telephone communication (for a full history of communication please see **Appendix E**).

<u>Interface requirements</u>:

Since the ambiguity of the current forms is one of the primary concerns of the current system, the interface of the new system must be straightforward, simple, and provide remote LAS technicians with an easy to use and thorough new interface. The system interface can be divided into separate parts for the users at the remote locations and the accountants at the central accounting office.

Technicians at remote locations:
This interface will appear much like a spreadsheet where the technicians will be able to enter in the primary data into the appropriate places. It is very important that the spreadsheet be as thorough as possible without being confusing. It needs to contain all of the necessary fields and data areas while at the same time not overwhelming the users and confusing them. It is very important to keep in mind during the interface design that a big problem currently is the data being entered into the system at the beginning stages. The interface needs to ensure that the maximum amount of accurate information can be entered in the simplest manner. This is by no means an easy task and heavy interaction with technicians and the accounting office during the design stage must occur to ensure that the interface

optimize efficiency at both ends and that a system is not put into place that does not in fact fix the current problem.

Accountants at centralized office:
The billing information that is compiled by the system needs to be presented to the accountants in a pre-ordered, ready to print, manner. The accountant should not need to take large amounts of time and order the data, this is precisely what the new system is intended to do away with. Searches will be implemented by researcher as well as facility searches that would bring up the total costs incurred by the system and the census information. Bills must be generated by animal type and housing.

Performance Requirements:

Time Bounds:

Remote Users:
There are a total of 12 remote locations that would need access to the system. In the worst case scenario, all twelve locations would try and send information into the central office at the same time. Since data is entered into the system at the end of technicians shifts, this is worst case is technically possible if the shift end times are synchronized. However, since 12 is a relatively small number of connections, this should not prove to be a large problem.

Management:
It must also be taken into consideration that people in the management office or accounting office may want to access the identical information at the same time. This must be allowed for as well as the possibility that these offices are accessing this information at the same time as submissions into the system are being entered from remote locations.

Uptime:
The system needs to be running a full seven days a week. The system may be shut down for maintenance provided that these interruptions are scheduled and approved ahead of time. Currently a lot of system maintenance and down time occurs on the weekends at the Veterinary school.

Reliability:

Information can be backed up once a month to minimize the affects of a system failure should one occur. The data can be backed up onto CD. If it turns out that once a month is not often enough due to the amount of information being accumulated each week this may have to be updated to twice a month or even weekly. Updating more than once a week though would not be practical as these

updates should occur during system downtimes which will generally occur on weekends.

Security:

Access to the system should be limited to LAS remote locations and the accounting office. The remote labs probably need very little access beyond submission of data. However they will be permitted some access to census data just so that they can make sure it is correct. Otherwise primary access is restricted to those in the accounting office.

Security is becoming and increasing concern for LAS due to certain complaints by animal rights activists and groups who have attempted to break into the facilities in the past. The system must not be accessible anywhere outside of the LAS facilities. To ensure security, and in compliance with the current use of Windows NT servers by the Veterinary school, the high encryption service pack for NT will be used. This is a 128-bit encryption system.

Survivability:

As discussed above all data will be backed up regularly onto Compact Disc to guard against data loss in case of an emergency.

Operating Requirements:

Physical constraints:
The system will run off of a server located in the accounting office. Thus space will not be a big issue. Also the storage of the backup CDs will also be relatively space efficient.

Personnel:
Current staff members will be operating the new system. This will require that all staff undergo a training session. Also LAS would like some sort of training manual to remain on hand at the sites of use of the system to allow for easier training of future staff members without the necessary inclusion of a systems specialist. There will also have to be a position in the accounting office (either new or just an old one with added responsibility) who is in charge of maintaining the system. For example, somebody needs to oversee the process of backing up the data.

All major changes and upgrades to the system will require the skill of a systems manager, probably a position that does not currently exist.

Lifecycle Requirements:

Quality of Design:

Since the design uses relatively standard web-based programming techniques, implementation of the design should not be that complex. It also allows for relatively easy maintenance as the technology develops.

Ideally Cornell would like to see the new system be functional for at least five years. After that they would prefer to be able to just add enhancements to the software or upgrade periodically rather than implement an entirely new system yet again. Due to the flexibility that the web-based solution offers, it does not seem as though this will be a problem.

Limits on Development:

LAS would like to see the system in place within nine months from the present.

Economic Requirements:

Cornell only wants to spend between 50 and 100 thousand dollars for initial development, with 100 as the absolute maximum (see **Appendix E.8**). They are willing to spend, however, 10-20 thousand a year after that. Once again, 20 is the absolute maximum.

Human Factors:

In the development of the new system, obviously the user-friendliness is very important. Part of the reason the new system is being developed is due to ambiguities currently plaguing the system and the forms which technicians use to send data to the accounting office. The new system must have a very simple design that very clearly lays out the required fields and options. Also, it must allow for a comment section where the tech may add points which they find important and my not appear explicitly on the predetermined form. (See **Appendix E.4**, and email from Mr. Pollard.)

Testability:

Since the change between the new system and old way is so dramatic, a small period of adjustment will be required. In this time both systems will be in place. Both the paper forms will be submitted as well as the new electronic versions. This is for a couple of reasons. Firstly, it will allow a more gradual change to the new system and also when issues arise they can be addressed and components added to the new system as necessary. Once the technicians are comfortable with

the system, and confident that it is working well, the old system will be phased out.

As time moves on, there will have to be evaluation periods where users can comment and supervisors can ensure that the system is still upholding the requirements that it was set out to.

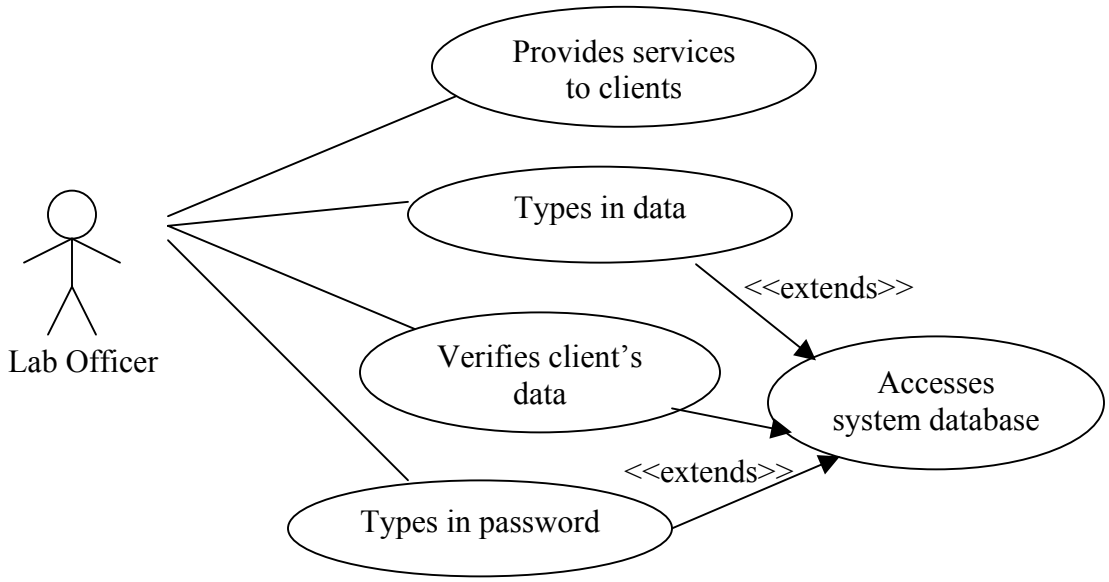
APPENDIX A – USE CASE DIAGRAMS

Actors Involved:

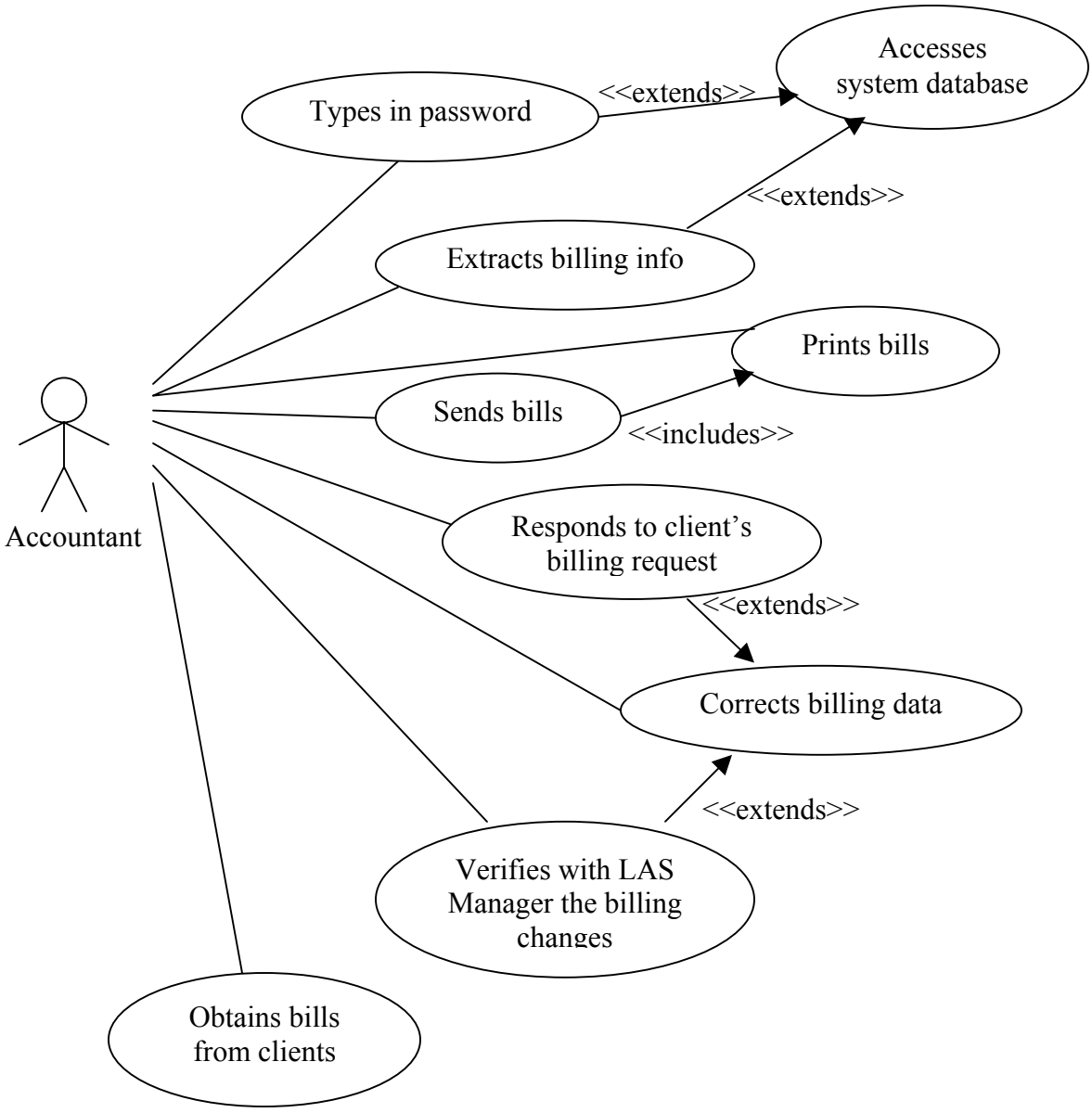
1. Lab Technician
2. Accountant
3. LAS Manager
4. Client (Researcher/Professor)
5. Software Manager
6. Database Administrator
7. Printer

Based on the Actors provided we can come up with a number of use cases to monitor the activities within the system. Due to numerous actors and use cases in the system we will first illustrate only these use cases which are involved with each actor listed above. Then we will create a bigger picture diagram to demonstrate the dependencies between the use cases and that some use cases are in used by more than one actor.

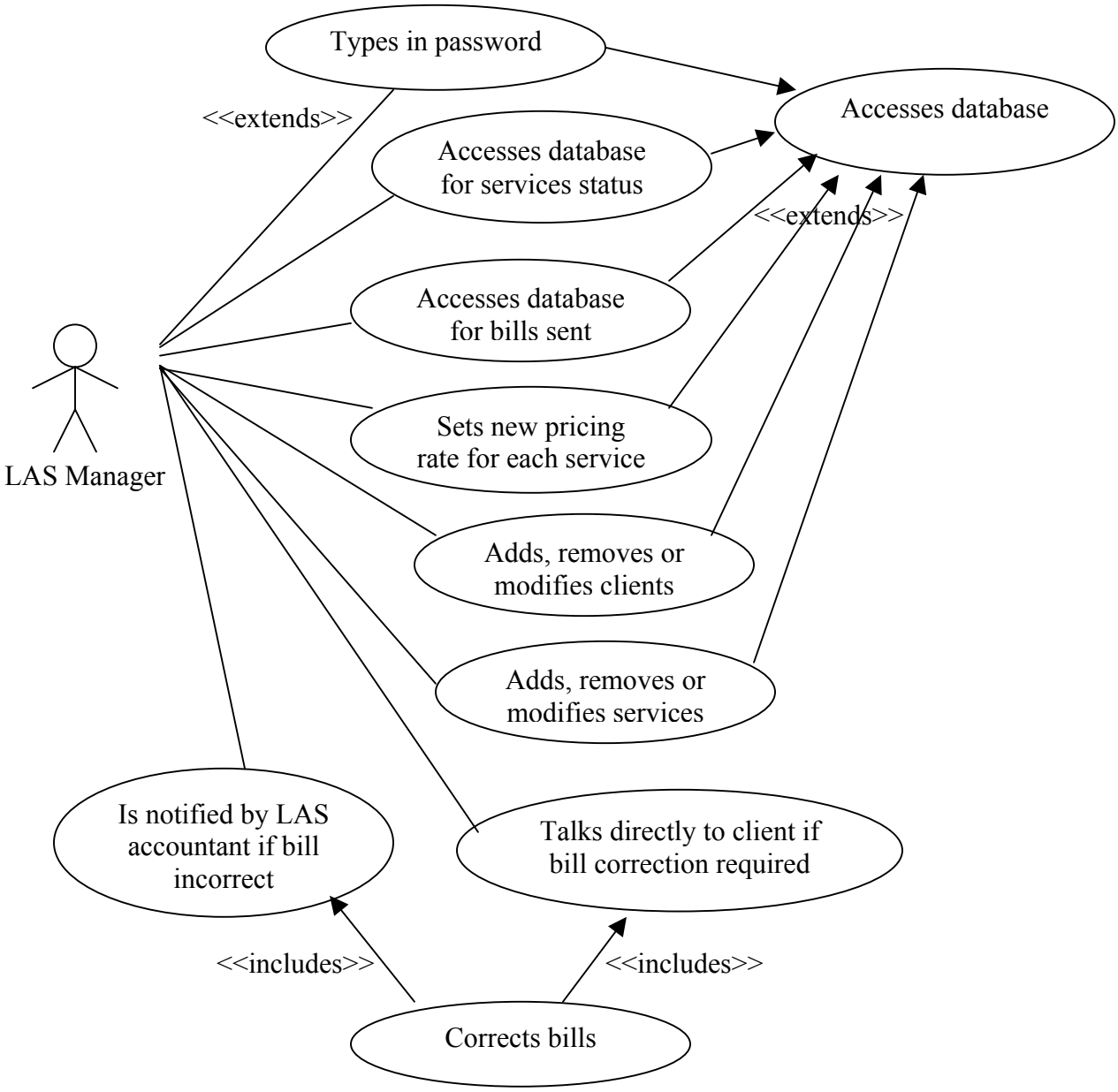
Use Cases for a Lab Technician



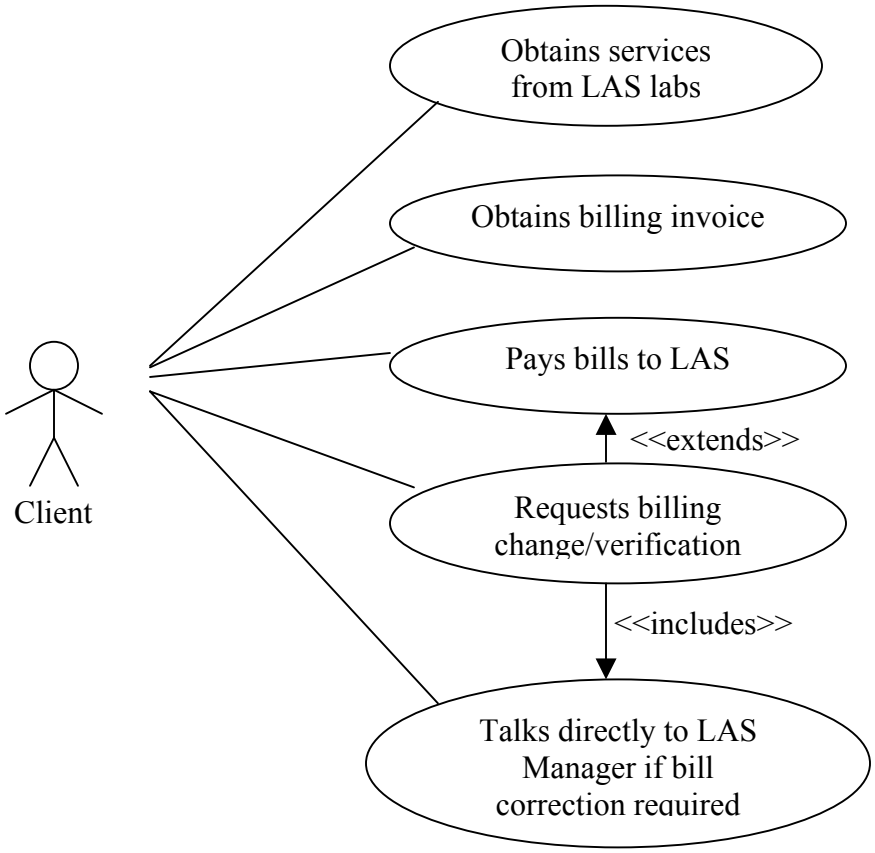
Use Cases for an Accountant



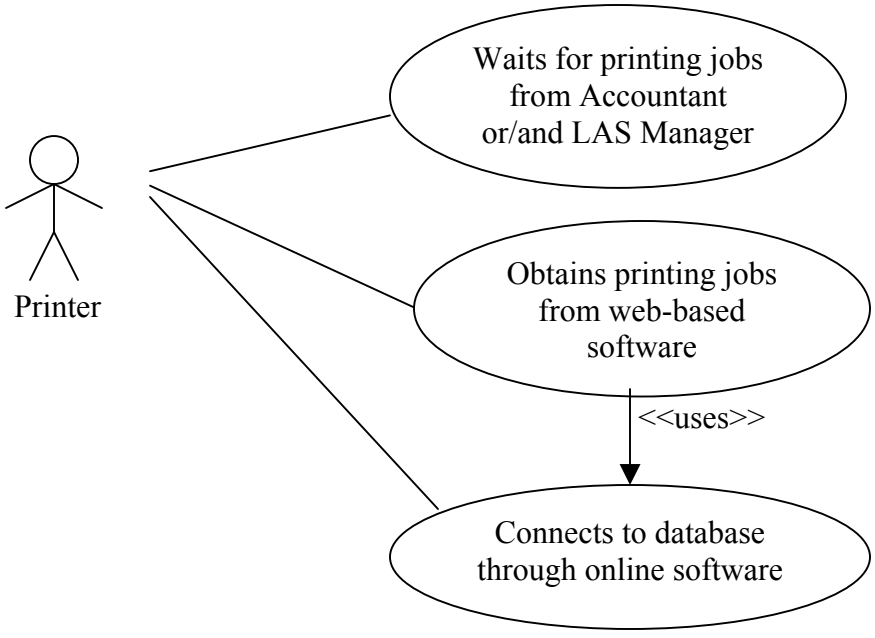
Use Cases for a LAS Manager



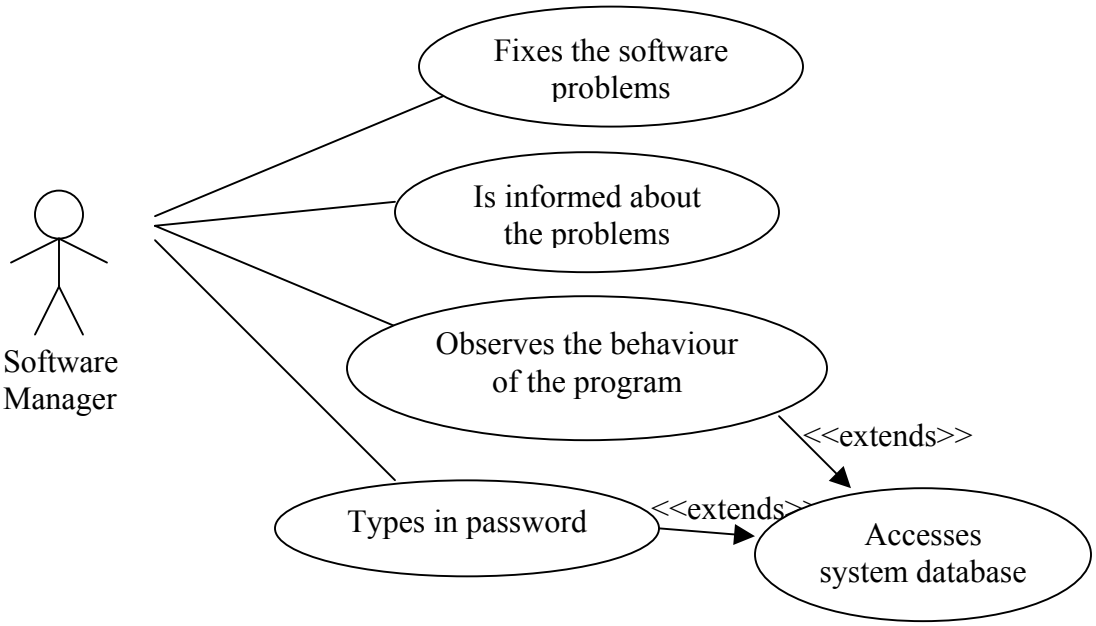
Use Cases for a Client



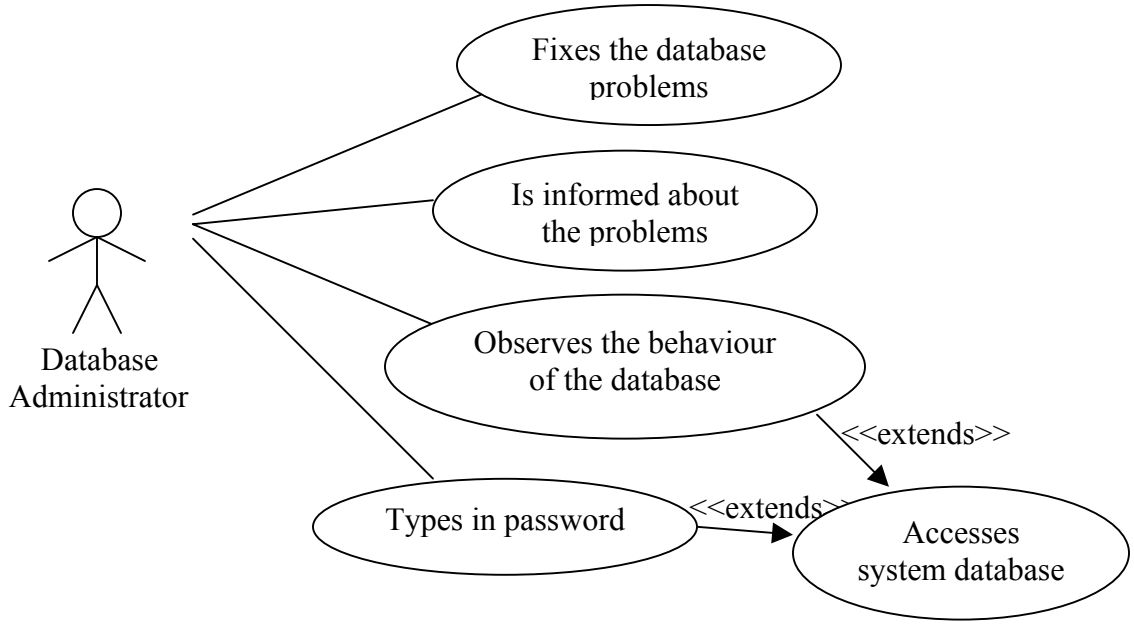
Use Cases for a Printer



Use Cases for a Software Manager



Use Cases for a Database Administrator



Bigger Picture Use Cases Diagram (Simplified)

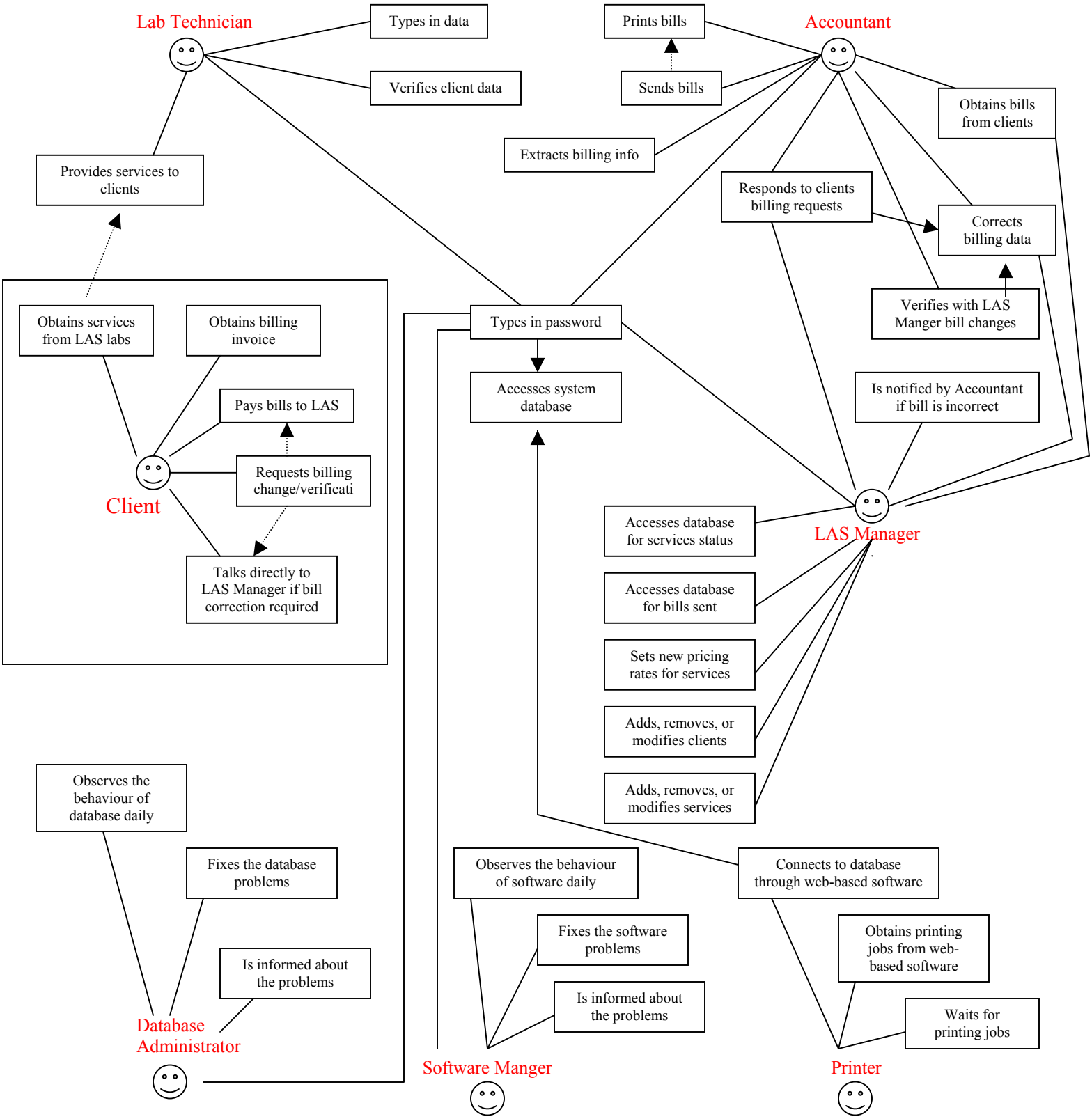
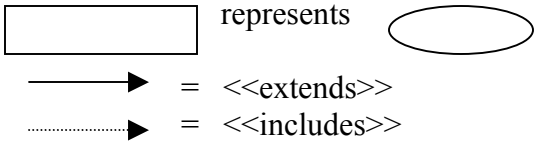


Diagram Legend:

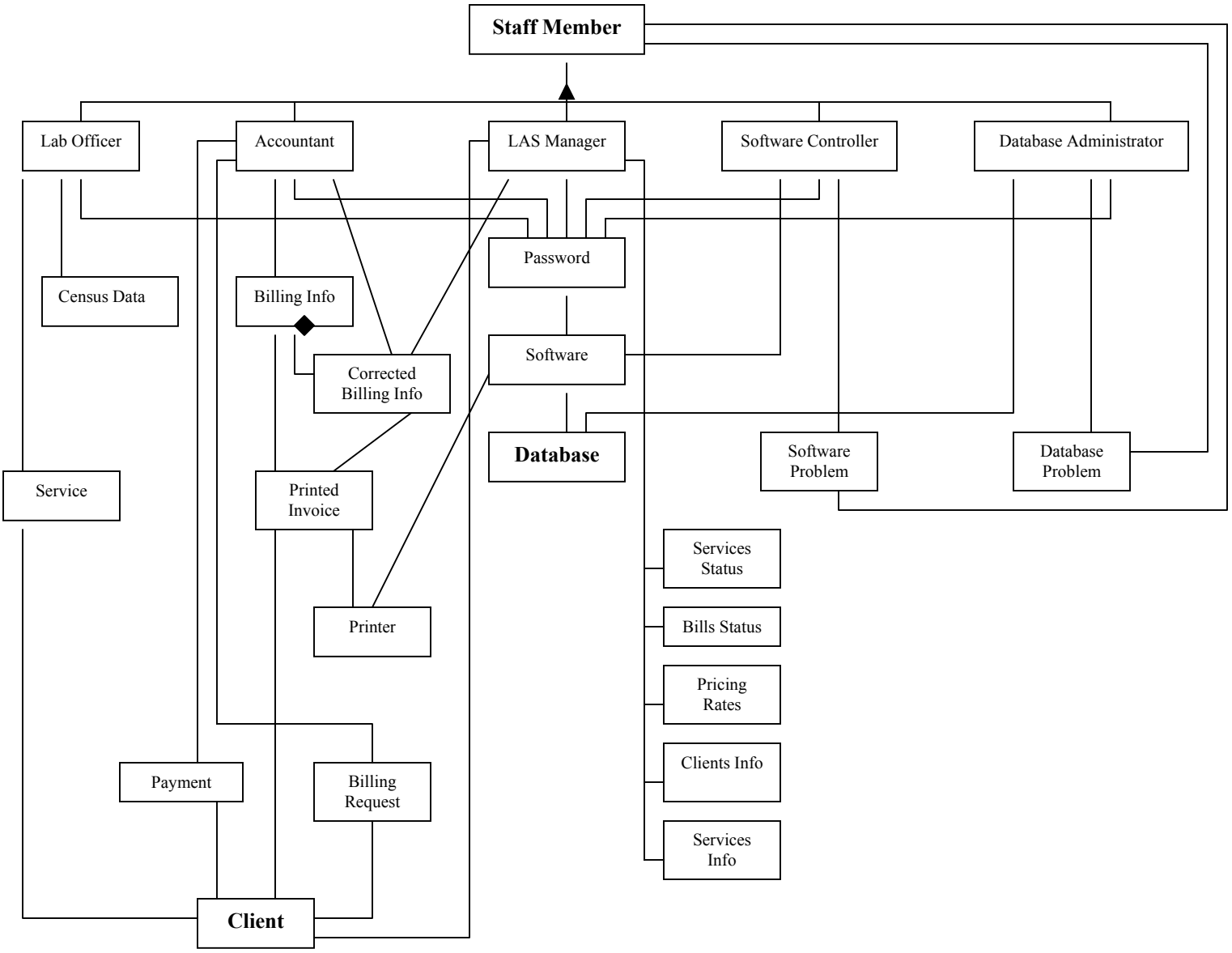


Use Case Descriptions

Use Case	Use Case Description
Type in password	Each user of the web-based software is required to enter his/her password before accessing the database information. Based on the password and the privileges assigned, the user will be connected to the specific interface of the software.
Access system database	Upon typing in the password the user will be connected to the database and will be allowed to access the information present there.
Type in data	Lab Technicians will be required to type in a daily census data. This will consist of the type and the quantity of services provided to each client.
Verify client's data	Lab Technicians will be allowed to re-enter the database at any time to verify if the data was entered correctly by them.
Provide services to clients	Lab Technicians are direclty connected to the clients by providing daily services to them.
Print bills	Once a month the accountants gather all billing data for each client and printed it out.
Send bills	After the bills are printed they are sent to the clients.
Obtain bills from client	If the client is not satisfied with the status of the bills received they send the bills back to LAS and the accountants obtain those bills for verification.
Respond to clients' billing requests	Accountants as well as the LAS Manager respond to the clients if the request for re-billing is obtained
Verify with LAS Manager the billing changes	In case the client requests billing verification, the accountants cannot change the billing information by themselves. They have to verify with management first what billing changes will be included.
Extract billing info	The accountants extract the billing information from the database before printing and sending them.
Correct billing data	Accountants and LAS Manger are responsible for correcting the billing data if it is incorrect.
Is notified by Accountant if bill incorrect	LAS Manager is notified by the accountants that the clients requested bill verification.
Access database for	LAS Manager manages and checks on daily basis the

Services status	information present in the system including Services status entered by the lab technicians.
Access database for Bills sent	LAS Manager checks daily the information present in the system including status of bills sent.
Set new pricing rates for each service	Each day the manager will check the pricing rates used for the services and will set and enter new rates if required. The software will have a capability of estimating the approximate rates based on the net incomes from services and LAS expenses.
Add, remove, or modify clients	LAS Manager's responsibility will be to manage the client's information in the database. He will modify this information accordingly.
Add, remove, or modify services	LAS Manager's responsibility will be to manage the information of the services provided in the database. He will modify this information accordingly.
Talk directly to client if bill correction required	Upon being notified that the bills received by the client need updating the manager will talk directly to the client.
Obtain services from LAS	Clients obtain services from the LAS labs on daily basis.
Obtain billing invoice	Each month the client will receive a bill invoice from LAS for the services obtained.
Pay bills to LAS	Upon obtaining the bill, the client has an option to pay the bill.
Request billing change/verification	Upon obtaining the bill, the client has an option to request form LAS accounting unit re-billing or billing verification.
Talks directly to LAS Manager if bill correction required	The LAS Manager will talk directly to the client to update/verify changes to the billed information.
Fix the software problems	The Software Controller will maintain the Web-based software. He will fix the software problems if any will occur.
Is informed about the software problems	Software Controller occasionally will be informed by other staff members of the existing software problems.
Observes the behaviour of the software	Software Controller's responsibility will be to check the software on daily basis by reading the error logs.
Fix the database problems	The Database Administrator will fix any problems occurring on the database.
Is informed about the database problems	Other staff members will inform the Database Administrator about the encountered problems.
Observe the behaviour of the database	Database Administrator will observe the database on daily basis by reading error log files.
Wait for printing jobs from Accountant and/or LAS Manager	The printer object will wait for the printing jobs assigned by the accountants or LAS Manager.
Obtain printing jobs from web-based software	Upon being assigned the printing job, the Printer will obtain the extracted information form the web-based software.

APPENDIX B – CLASS DIAGRAMS



NOTE TO THE MARKER: The detail and completed diagram is presented in the hardcopy version of the assignment. The hardcopy includes the names of each assignment, rules, and multiplicites.

Data Dictionary for the Class Diagram

This section of Appendix B has a purpose to provide a detail description for each class. This will include the purpose of each class in the system, the decription of the operations and attributes, the types assigned to the operations and attributes.

Class: **Staff Member**
Description: This class describes all the members of the LAS department at Cornell University.
Private Attributes:
Name:String
Represents the first and last name of the staff member
StaffID:Integer
Represents a 10-digit ID number assigned to each member of the university
StartDate: Date
The date the member started working for LAS
Password:Password
Each member has a specific password assigned to them
Public Operations:
GetName(StaffID:Integer):String
Returns the name of the staff member based on his/her id
GetID(Name:String):Integer
Returns the Id number of the specific member based on his/her name
GetDate(StaffID:Integer):Date
Returns the start date of the specific member's employment at LAS based on his/her staff id
TypeInPassword(Password:Password):BOOL
This operation describes the process of typing in the password to the web-based software. If the StaffMember is accessed to database TRUE is returned

Class: **Lab Officer**
Description: This class describes a Lab Officer member of the LAS department at Cornell University. Lab Officer is responsible for typing in census data on daily basis and providing services to the clients. This class is a child class of **StaffMember** class.
Inherited Private Attributes:
Name:String
StaffID:Integer
StartDate: Date
Password:Password
Public Attributes:
LabLocation:String
The name of the remote lab location to which this lab officer is assigned
Inherited Public Operations:
GetName(StaffID:Integer):String
GetID(Name:String):Integer
GetDate(StaffID:Integer):Date
TypeInPassword(Password:Password):BOOL
Public Operations:
GetLabLocation(StaffID:Integer):String
Returns the remote lab location for this lab officer works
TypeInData(CensusData:Data)
This is an operation, which demonstrates that the lab officer types in the CensusData (look for class description below) to the database.
VerifyData(CensusData:Data):BOOL
This operation has a purpose to verify the existing CensusData. If lab technician needs to change the data, the data will be changed and this operation will return TRUE. If the data does not need to be changed FALSE will be returned

Class: **Accountant**

Description: This class describes an Accountant member of the LAS department at Cornell University. LAS Accountant is responsible for generating billing information, sending bills and responding to clients' billing enquiries. This class is a child class of **StaffMember** class.

Inherited Private Attributes:

Name:String
StaffID:Integer
StartDate: Date
Password:Password

Inherited Public Operations:

GetName(StaffID:Integer):String
GetID(Name:String):Integer
GetDate(StaffID:Integer):Date
TypeInPassword(Password:Password):BOOL

Public Operations:

ReceivePayment(Payment:Data):BOOL
This operation describes that a specific payment was obtained from the client. Additionally if payment was received successfully TRUE is returned, if no FALSE returned.

SendBill(PrintedInvoice:Data):BOOL
This operation describes that a specific printed bill was sent to the client. Additionally if bill was sent successfully TRUE returned, if no FALSE returned.

GenerateBillingInfo(ClientID:Integer):BillingInfo:Data
This operation generates a BillingInfo data.

RespondToBillingRequest(BillingRequest:Data):BOOL
In case a client sends a billing request, the accountant is responsible to respond to this billing request. If responded successfully TRUE is returned; otherwise FALSE is returned

Class: **LAS Manager**

Description: This class describes a LAS Manager at Cornell University. LAS Manager is responsible for verifying billing information in case client complains about it, he talks directly to clients if billing problems occur. Additionally, LAS Manager accesses database daily to check and/or modify Services Status, Bills Status, Pricing Rates, Clients Info and Services Info. This class is a child class of **StaffMember** class.

Inherited Private Attributes:

Name:String
StaffID:Integer
StartDate: Date
Password:Password

Inherited Public Operations:

GetName(StaffID:Integer):String
GetID(Name:String):Integer
GetDate(StaffID:Integer):Date
TypeInPassword(Password:Password):BOOL

Public Operations:

VerifyBillingChanges(CorrectedBillingInfo:Data):BOOL
In case a client sends a billing request, the accountant is responsible to respond to this billing request. Changed billing information is sent to the LAS Manger and he verifies it. If verification is successful TRUE is returned; otherwise FALSE is returned.

TalkDirectlyToClient(BillingRequest:Data):BOOL
In case there is a billing request sent from the client, the LAS Manger has to talk to the client to obtain the detail information about the request and to get his approval for the changes. If the client approves the changes the function returns TRUE; otherwise FALSE is returned.

ModifyServiceStatus(ServicesStatus:Data):BOOL
On daily basis Manager has to check ServiceStatus; if information is changed TRUE returned; otherwise FALSE;

ModifyBillsStatus(BillsStatus:Data):BOOL
On daily basis Manager has to check BillsStatus; if information is changed TRUE returned; otherwise FALSE;
ModifyPricingRates(PricingRates:Data):BOOL
On daily basis Manager has to check PricingRates; if information is changed TRUE returned; otherwise FALSE;
ModifyClientsInfo(ClientsInfo:Data):BOOL
Once in a while the LAS Manager has to modify the ClientsInfo by adding, removing or changing clients' data; if information is changed TRUE returned; otherwise FALSE;
ModifyServicesInfo(ServicesInfo:Data):BOOL
Once in a while the LAS Manager has to modify the ServicesInfo by adding, removing or changing services provided by the remote labs; if information is changed TRUE returned; otherwise FALSE;

Class: **Software Controller**
Description: This class describes a Software Controller member of the LAS department at Cornell University. Software Controller's responsibilities include maintaining the web-based software and fixing software problems if any occur. He checks a daily error log for problems and is notified of problems by other staff members. This class is a child class of **StaffMember** class.
Inherited Private Attributes:
Name:String
StaffID:Integer
StartDate: Date
Password:Password
Inherited Public Operations:
GetName(StaffID:Integer):String
GetID(Name:String):Integer
GetDate(StaffID:Integer):Date
TypeInPassword(Password:Password):BOOL
Public Operations:
FixSoftwareProblem(SoftwareProblem:Data):BOOL
In case some problem exists on the web-based software the Software Controller fixes it. If the fix was successful TRUE will be returned; if the operation was not successful FALSE will be returned.
MaintainSoftware(Software:Program)
This operation describes a general procedure of maintaining a web-based software. This will include observing the behaviour of the software and correcting some issues of the software.
ReadErrorLogFile(LogFile:Data)
This a general procedure for reading an error log file generated by the software.
IsNotifiedAboutProblem(SoftwareProblem:Data):BOOL
This procedure is used when the Software Controller is notified by other StaffMember of the problems encountered on the software. TRUE if Software Controller was notified; FALSE otherwise.

Class: **Database Administrator**
Description: This class describes a Database Administrator member of the LAS department at Cornell University. His responsibilities include maintaining the main database and fixing database problems if any occur. He checks a daily error log for problems and is notified of problems by other staff members. This class is a child class of **StaffMember** class.
Inherited Private Attributes:
Name:String
StaffID:Integer
StartDate: Date
Password:Password
Inherited Public Operations:
GetName(StaffID:Integer):String
GetID(Name:String):Integer
GetDate(StaffID:Integer):Date
TypeInPassword(Password:Password):BOOL
Public Operations:
FixDatabaseProblem(DatabaseProblem:Data):BOOL
In case some problem exists on the main database the Database Administrator fixes it. If the fix was successful TRUE will be returned; if the operation was not successful FALSE will be returned.
MaintainDatabase(Database:Program)
This operation describes a general procedure of maintaining a main database. This will include observing the behaviour of the database and correcting some issues of the database.
ReadErrorLogFile(LogFile:Data)
This a general procedure for reading an error log file generated by the database.
IsNotifiedAboutProblem(DatbaseProblem:Data):BOOL
This procedure is used when the Database Administrator is notified by other StaffMember of the problems encountered on the main database. TRUE if Database Administrator was notified; FALSE otherwise.

Class: **Census Data**
Description: This class describes the census data entered to the database by the Lab Officer
Private Attributes:
DataType:String
Represents the type of data entered. This could be the quantity of services provided, the type of animal used, which client has used a service, etc.
Public Operations:
GetGeneralCensusData(ClientID:Integer):CensusData
Upon entering the ClientID information this operation will provide the census data entered for this client.
GetSpecificCensusData(ClientID:Integer, DataType:String):CensusData
Upon entering the ClientID and the CensusType required, the operation will return all the CensusData based on the Census Type entered.

Class: **Service**
Description: This class describes the services provided to clients by the Lab Officer
Private Attributes:
ServiceType:String
Represents the type of service provided to the client.
NumberOfServicesAvailable:Integer
This is the total number of services available in the LAS lab.
Public Operations:
GetGeneralServiceInformation(ClientID:Integer):Service
Upon entering the ClientID information this operation will provide the list of services provided to this client.
GetSpecificServiceInfo(ClientID:Integer, ServiceType:String):Service
Upon entering the ClientID and the ServiceType provide, the operation will return the list of services of type ServiceType provided to the client.

Class: **Billing Info**
Description: This class describes the billing information generated by the Accountant.
Private Attributes:

ClientName:String
The name of the client on the billing info.
ClientID:Integer
The ID of the client on the billing info.
AmountOwn:Money
The amount of money own by the client.
DetailBillingInfo:Data
Extra billing information.

Public Operations:

GeneratePrintedInvoice(ClientID:Integer):PrintedInvoice
This function generates the printed invoice based on the ClientID.
GetMoneyOwn(ClientID:Integer):AmountOwn
Returns the amount own by a specific client.
GetDetailBillingInfo(ClientID:Integer):DetailBillingInfo
Returns the detail billing information about this specific client.

Class: **Corrected Billing Info**
Description: This class describes the corrected billing information generated by the Accountant and verified by the LAS Manager. This is a child class of **BillingInfo**class.
Inherited Private Attributes:

ClientName:String
ClientID:Integer
AmountOwn:Money
DetailBillingInfo:Data

Public Attributes:

ChangedBillingInfo:Data
It represents the billing data that was changed

Inherited Public Operations:

GeneratePrintedInvoice(ClientID:Integer):PrintedInvoice
GetMoneyOwn(ClientID:Integer):AmountOwn
GetDetailBillingInfo(ClientID:Integer):DetailBillingInfo

Public Operations:

GetChangedBillingInfo(ClientID:Integer):ChangedBillingInfo
This operation returns the billing information that was changed by the accountant due to clients' request for bill change.

Class: **Printed Invoice**
Description: This class describes the printed bill. A bill is printed upon generating the billing info by the accountant and sending the print job from the we-based software to the printer.
Private Attributes:

ClientName:String
The name of the client on the printed bill.
ClientID:Integer
The ID of the client on the printed bill.
AmountOwn:Money
The amount of money displayed on the bill which is own by the client.
DetailBillingInfo:Data
Extra billing information on the printed bill.

Public Operations:

IsSentPrintedInvoice(ClientID:Integer):BOOL
This function is responsible for determining if the printed invoice was sent to the client. If operation successful TRUE returned; otherwise FALSE returned.

Class: **Printer**
Description: This class describes the printer object. Printer's responsibility is to print a Printed Invoice upon obtaining a corresponding printing job form the web-based software.
Private Attributes:

JobNumber:Integer
Each job assigned to the printer by the web-based software has its job number. Based on this number we can determine which job has been printed and which one has not.

Public Operations:

PrintBill(JobNumber:Integer):BOOL
This operation has a purpose to print a bill based on the JobNumber obtained from the web-based software.

Class: **Payment**
Description: This class describes the payment information sent by the client and obtained by the Accountant.
Private Attributes:

ClientName:String
The name of the client on the payment.
ClientID:Integer
The ID of the client on the payment.
AmountOwn:Money
The amount of money own by the client shown on the payment.
AmountAttached:Money
The actual amount returned with the payment.

Public Operations:

HasPaymentBeenReceived(ClientID:Integer):BOOL
Determines if the accountant has received the payment. If yes TRUE returned; FALSE otherwise.
IsFullPayment(AmountOwn:Money,AmountAttached:Money):Money
Compares the amount of money received with the amount of money own and returns the difference

Class: **Billing Request**
Description: This class describes the billing request sent by the client in case the client does not agree with the amount shown on the Printed Invoice.
Private Attributes:

ClientName:String
The name of the client on the billing request.
ClientID:Integer
The ID of the client on the billing request.
TypeOfRequest:String
This string represents the type of request sent. Some types of request include – wrong amount displayed, this amount has been already paid; the services shown on the bill are incorrect, etc.

Public Operations:

RequestWasReceived(ClientID:Integer):BOOL
This operation is used if the client sends the billing request. If accountant obtains the request TRUE is returned; FALSE otherwise.
RequestWasPrcessed(ClientID:Integer):BOOL
This operation is used when the billing request was processed by the accountant and was verified by the LAS Manager. TRUE if request was processed successfully; FALSE otherwise.
WhatTypeOfRequest(ClientID:Integer):TypeOfRequest
This operation returns the type of request sent by the specific client based on the ClientID.

Class: **Client**
Description: This class describes all the client using the services provided by the LAS department at Cornell University.
Private Attributes:

Name:String
Represents the first and last name of the client
ClientID:Integer
Represents a 10-digit ID number assigned to each LAS client
StartDate: Date
The date the client started using LAS service
ServicesReceived:ServiceType list
Represents the list of LAS services obtained by the client.

Public Operations:

GetName(ClientID:Integer):String
Returns the name of the client based on his/her id
GetID(Name:String):Integer
Returns the Id number of the client based on his/her name
GetDate(ClientID:Integer):Date
Returns the start date of the specific client based on his/her staff id
WhatServicesUsed(ClientID:Integer):ServicesReceived
Returns the list of services provided to this specific client
StatusOfPayment(ClientID:Integer):Money
This operation checks if the client owns any money to LAS accounting unit and whet that amount is.

Class: **Password**
Description: This class describes the password object, which is required to be entered before accessing the database.
Private Attributes:

Password:String
The actual Password required to access the software and database. Notice that each StaffMember has its own Password assigned to them.

Public Operations:

VerifyPassword(Password:Password):BOOL
This operation verifies if the password entered by the user is corresponding to the password assigned to him.

Class: **Software**
Description: This class describes the web-based software object. Each user is required to enter his/her password before accessing the software. The software object communicates directly with the main database object and its main role is to extract and send the data assigned by the user. Additionally software object sends print jobs and is maintained by the Software Controller.
Private Attributes:

InterfaceTypes:Interface
Software has 3 interface types access to which depends on the privileges assigned to the users.
ProgramData:Data
The data stored in the software, which can be accessed or changed by the user.

Public Operations:

ExtractData(ProgramData:Data):BOOL
This operation allows to access/extract the data from the software. If data extracted successfully TRUE returned; otherwise FALSE;
Access()
Allows an access to database

Class: **Database**
Description: This class describes the database object. Database communicates directly and exchanged data with the Software object depending on the queries sent by the Software.
Private Attributes:
DatabaseData:Data
This is the main database data which can be accessed by the Software.
Public Operations:
ExtractData(DatabaseData:Data):BOOL
This operation allows the Software to access/extract the data from the Database. If data extracted successfully TRUE returned; otherwise FALSE;

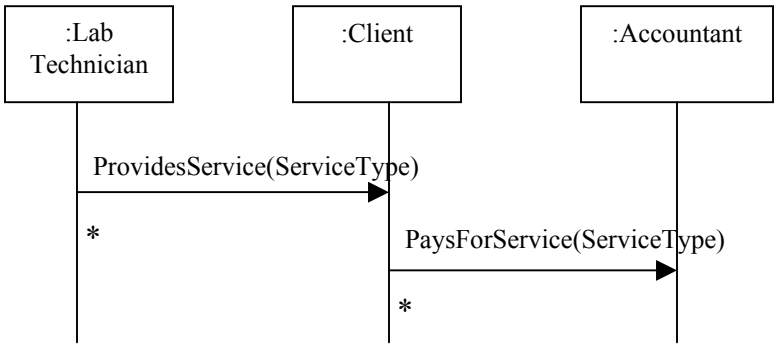
Class: **Software Problem**
Description: This class describes the software problems that could occur on the web-based software. These problems are either reported by any StaffMember or uncovered by the Software Controller during the course of maintaining the software.
Private Attributes:
ProblemNumber:Integer
Each problem reported has a number assigned to it.
ProblemType:String
The problems on the software could be of many types. This attribute determines what problem type this Software Problem has.
AssignedBy:StaffMember
The staff member who has reported the problem.
Public Operations:
ReportProblem(AssignedBy:StaffMember):ProblemNumber
This operation has a purpose to report a software problem by a staff member. If problem is reported it is assigned a ProblemNumber.
AssignProblemType(ProblemNumber:Integer)ProblemType
This operation assigns a type of problem to the specific ProblemNumber.

Class: **Database Problem**
Description: This class describes the database problems that could occur on the main database. These problems are either reported by any StaffMember or uncovered by the Database Administrator during the course of maintaining the database.
Private Attributes:
ProblemNumber:Integer
Each problem reported has a number assigned to it.
ProblemType:String
The problems on the database could be of many types. This attribute determines what problem type this Database Problem has.
AssignedBy:StaffMember
The staff member who has reported the problem.
Public Operations:
ReportProblem(AssignedBy:StaffMember):ProblemNumber
This operation has a purpose to report a database problem by a staff member. If problem is reported it is assigned a ProblemNumber.
AssignProblemType(ProblemNumber:Integer)ProblemType
This operation assigns a type of problem to the specific ProblemNumber.

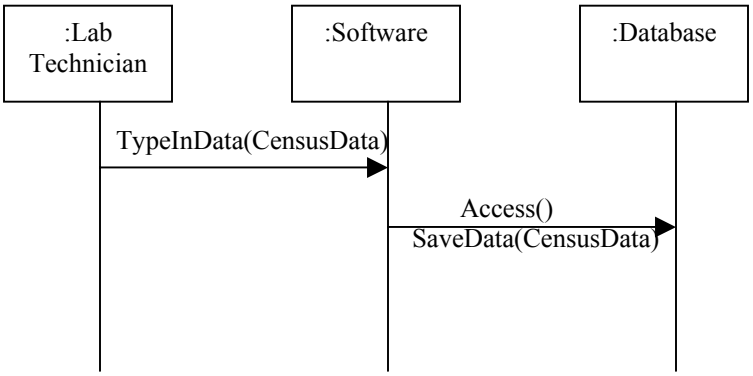
APPENDIX C - SEQUENCE DIAGRAMS

We have identified a number of sequence diagrams in our system, which allow to monitor the intractions between the system objects in a time sequence. Here are our digrams:

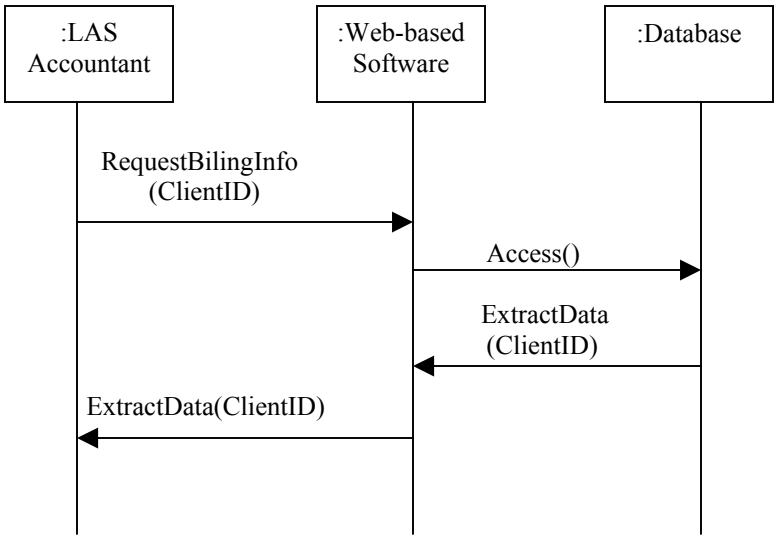
Providing Services to Client



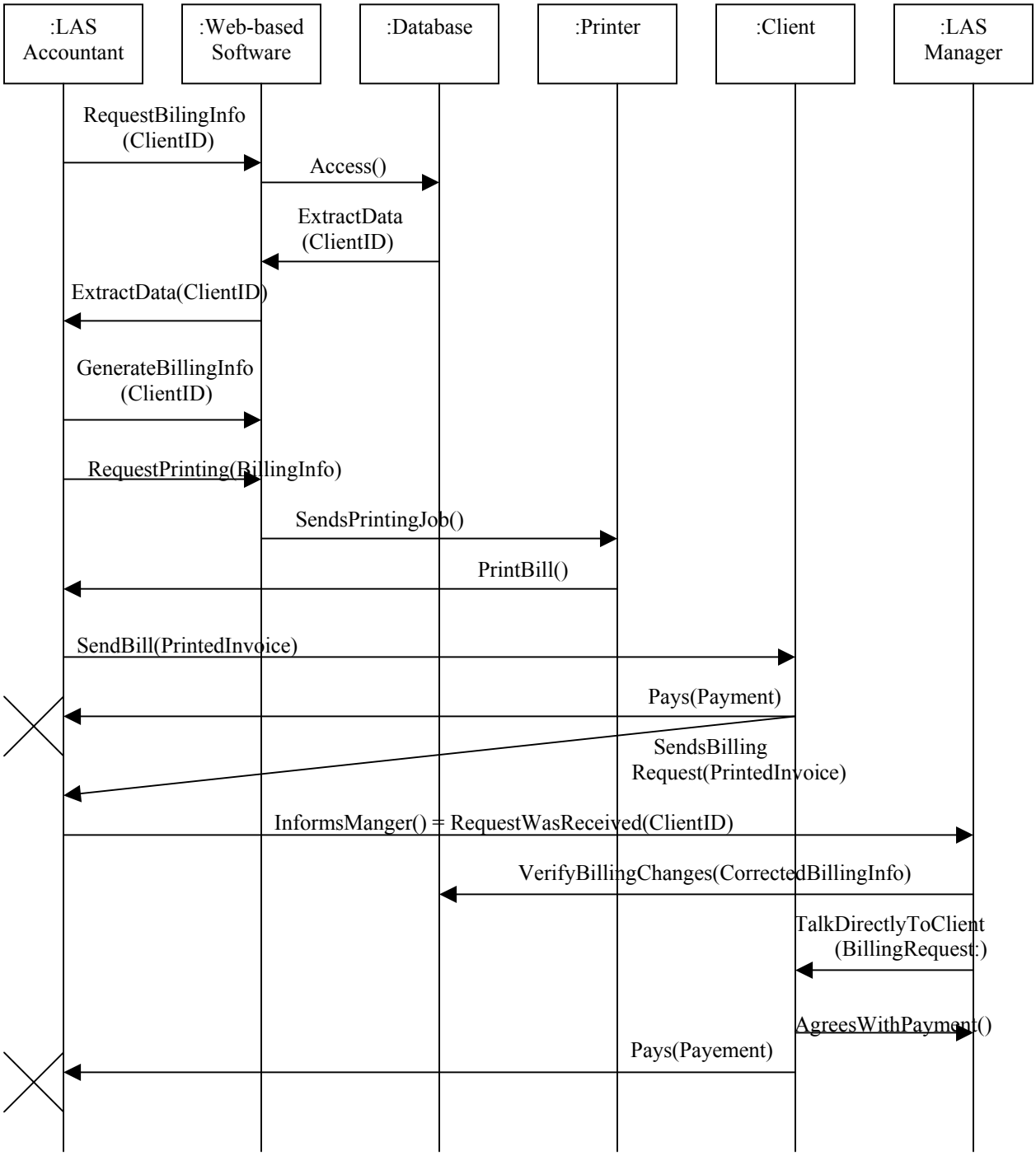
Census Data Entering To Database Sequence Diagram



Bill Information Extracting From Database Sequence Diagram

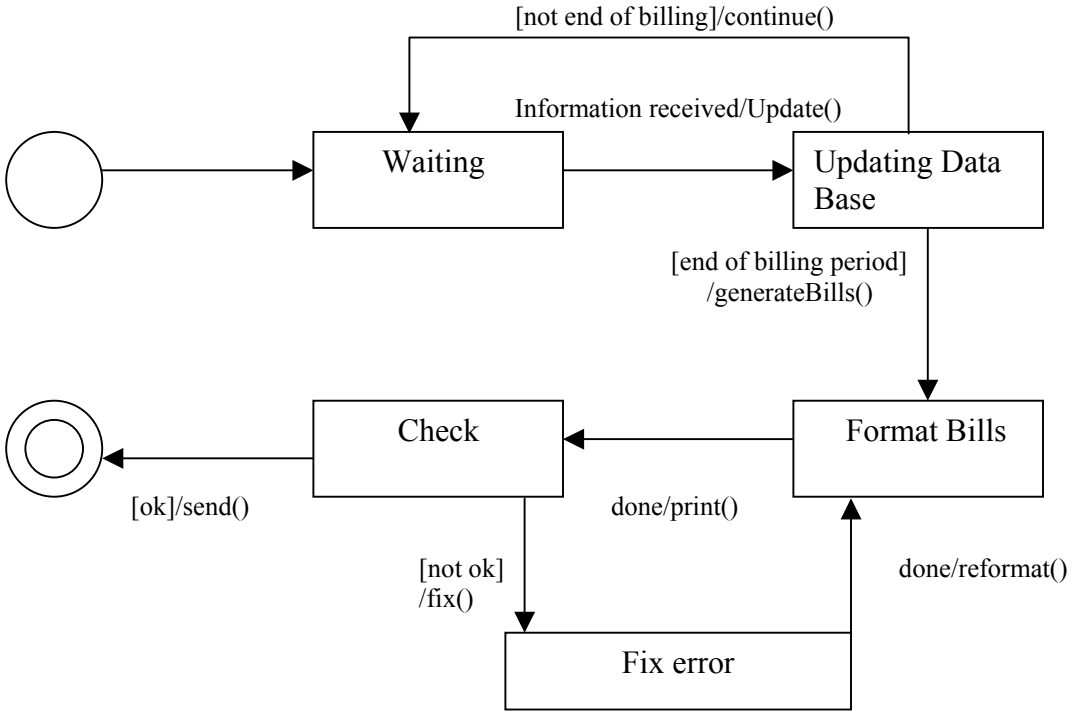


Bill Processing Sequence Diagram

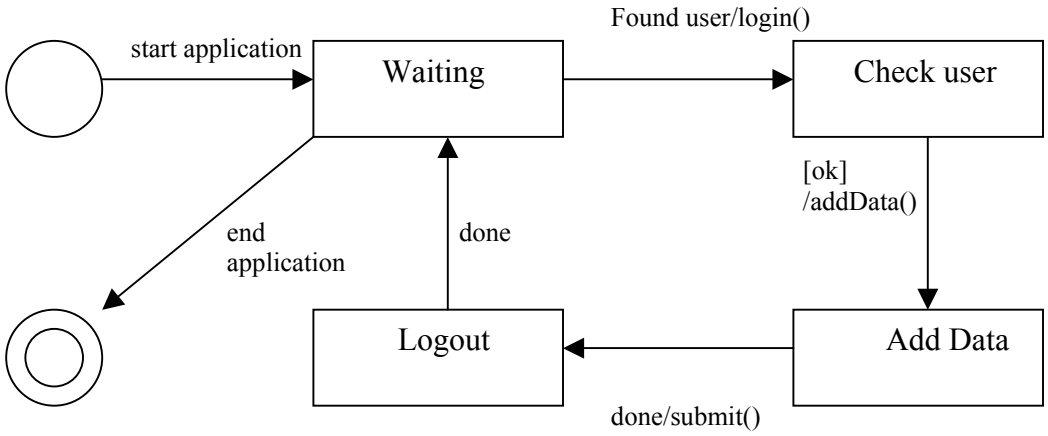


APPENDIX D - STATE DIAGRAMS

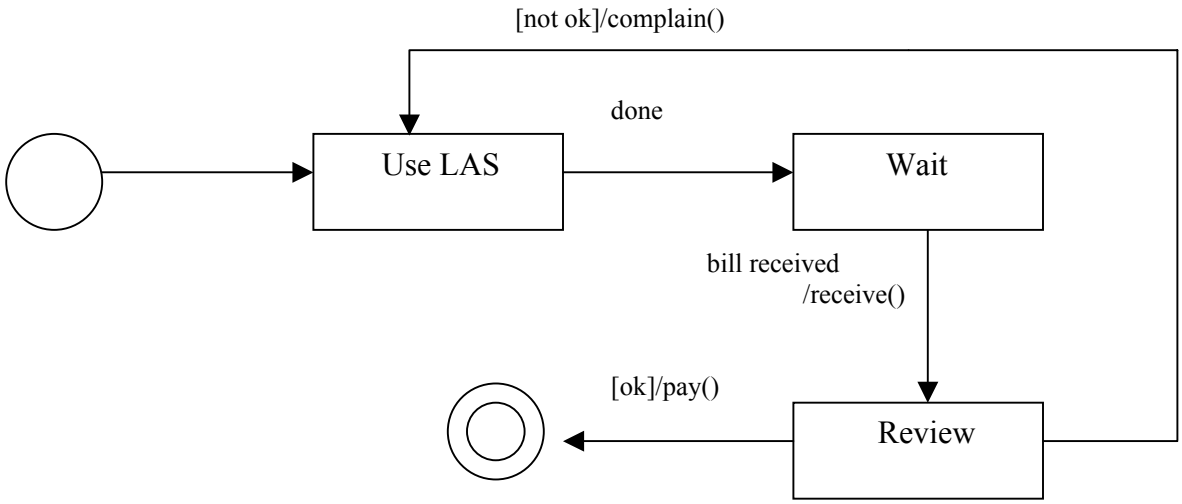
D.1-Accounting Office



D.2- Remote Location



D.3- Researcher



Appendix E

E.1

CONTACT INFORMATION:

All contact with Cornell and LAS was done through these two contacts (a detailed report of all contact follows):

Timothy L. Pollard
VM Clinical Prgs & Prof Svcs
NetID tlp7
Email Address tlp7@cornell.edu
Campus Address: C3-505 Vet Medical Center Campus Phn 607-253-3946
FAX 607-253-3933
Type: staff
Department: VM Clinical Prgs & Prof Svcs
Working Title: Assistant Director - Financial Operations

Robert O. Gilbert,
BVSc, MMedVet, director
Box 28, College of Veterinary Medicine
Cornell University
Ithaca, New York 14853-6401
phone: (607) 253-3472, fax: (607) 253-3440

Appendix E.2

January 25, 2002

Hi Dave.
I spoke to some of the folks who have been involved in our Information System pain of the past few years. As I said, we paid a consultant $50,000 for an initial study and have poured a lot of money in since. The scope is huge (for the Diagnostic Laboratory and Hospital). They thought that even the pharmacy piece I had thought of was too big – it interfaces with so many other sections.
All is not lost, however. We have a brand new and isolated problem.
Here's the deal. I have just assumed responsibility for the Laboratory Animal Services component of the University. We have hired a new Director, and we are in the process of sorting out a significant mess. We need to establish new systems for a lot of activity. In a nutshell, LAS provides services to investigators (researchers) or teachers who use animals. They maintain animals at several different sites. These involve many different species of animals: mice, rats, rabbits, voles, fish, sheep, birds, cattle, dogs, cats, horses, etc. They need to bill individual investigators or teachers each month. At present they do not even have an accurate daily census of animals. A technician caring for the animals will (or not) record the number of animals of a given sort taken care of. She turns in those slips to someone who physically enters the data (on paper) and consolidates the reports each month to generate bills.
We need to bill by animal type and by housing (some are in special housing -- e.g. isolation). Daily rates are established in advance for each species.
Ideally, we would coordinate this system with other aspects of the management – equipment inventory, food inventory and ordering, bedding, etc. Also, it would be valuable, from a cost accounting point of view, to be able to break down the time it took to take care of each category of animal. Cost accounting is important because this is (by law) a break even operation – we are not allowed to lose or make money. The list could go on.
This is a real problem in real time. We actually do need an answer and are working toward one.
Would it work for you?
What do you need to get started?

Dr. Gilbert

Appendix E.3

January 28, 2002

Hi Dave.

Got your phone message. I will be in and out. Tim Pollard is an accountant who has to deal with LAS. He has given me a package of paper that I will send by Fed-Ex. I have not really had a chance to look at it yet. I'll be happy to answer any questions that I can. Feel free to ask Tim too.

Dr. Gilbert

Appendix E.4

January 28, 2002

Hi David,

We just took over the accounting for the Laboratory Animal Services unit at the Vet School. This dept provides all the care and maintenance for animals used for research. Researchers are billed on a monthly basis for the services provided. The census data is compiled in a very manual manner, then forwarded to our office. We keypunch the handwritten forms into an Access database containing rates by species, researcher name, and general ledger charging data. When keying is complete, the data is exported to Excel to verify the rates charged by species are correct. Then the invoices (2 sets) are printed. One set mailed to the accounting office of the researcher, the other set matched with the handwritten forms and filed. Customers have 1 week to respond with questions, changes in general ledger attributes, etc. At the end of the week, the data is summarized in the database and transformed into a format that can be uploaded to the G/L. The final step is to create accounting entries to transfer small amounts of the income from that sales entry to offset a prior year deficit and to build an equipment reserve. This cycle is repeated every month.

Problems:

1) Handwritten notes are used for billing - hard to read, data has to be re-keyed into billing system by accounting staff who have no knowledge (yet) of how each lab area works. Form not set up to capture enough data in some cases. Time consuming to do all keying by one person - about 5 man-days per month, lots of opportunity for error.

2) No good way to track "other" services provided that need to be billed. Ex - special supplies, purchase of animals, labor to move, crate, or ship animals, snowplow work, lawn mowing, etc. Being done manually from memory now - often billable items are lost (I think).

3) Needs exist to track other types of data; protocols for animal use, health profiles of workers, costing data to help determine billing rates. Existing protocol system is automated, but at least 15 years old.

Environment:

1) Computer skills of animal care workers probably not very good. Expect each area supervisor be better, but not sure how much better.

Big transition/training issues. Some locations are off campus - need web front end to access system from remote locations. Need strong firewalls - data needs to be secure for a variety of reasons.

2) Need to improve financial position of the unit.

3) Unit functions under University Service Center policy. Rates must be calculated to break even each year, with allowance for prior year surplus/deficit and funding of capital equipment reserve. Currently all calculations performed with little help from census system.

4) Commercial systems are available. Need to identify best fit, do gap analysis to determine any customization required. Then be sure we can afford cost of purchase plus customization. Site visits planned to universities using systems under consideration.

That's about all I can think of right now. We have just started discussing a new system this month, so this is a real time project. I will put together a package of some non-confidential data to ship or fax to you. Let me know how you want to proceed from here.

Tim

Appendix E.5

January 29, 2002

Hi Dave,

The package contains a sample of the monthly census report (handwritten), and a sample of the monthly invoice we create for one animal containment area. Also, a magazine article that compares several of the systems we are considering, as well as sales material from some of the companies we have contacted. What we don't have yet is a list of our specific needs/unique requirements. I'm sure we will create one soon; I will forward it. In the meantime, feel free to contact me with any questions you have. E-mail is probably best - I'm out of the office often for meetings.

Good luck,

Tim

-----Original Message-----
From: dave gilbert [mailto:david.gilbert@utoronto.ca]
Sent: Monday, January 28, 2002 4:47 PM
To: Timothy L. Pollard
Subject: Re: Systems Project

Tim,

Thank you very much for your help. The outline that you provided in your earlier e-mail was very helpful and this project seems to be exactly what I am looking to study.

I spoke briefly with Dr. Gilbert and he mentioned that you had given him some information to send up to me. Once I receive the packet and look over it, would it be possible for me to contact you further with some follow up questions?

Thank you very much for your time and efforts in helping me with this project.

Dave

Appendix E.6

January 31, 2002

Hi Dave,

Here is a summary of the current year budget for LAS. I don't have a quick breakout of the non-personnel expenses. But the vast majority would be supplies (animal feed, cleaning supplies, etc). Other categories would be services, some equipment purchases, travel, and transportation. Also included would be depreciation on existing equipment (maybe 75K), and deficit recovery (maybe 50K of the 200K deficit).

Please keep this as confidential as possible.

Tim

Expenses

Exempt Salaries	225,250
Non-Exempt Salaries	1,135,168
Temp Salaries	54,460
Salary Improvement Prog	71,437
Non-Personnel	749,395
Total Expenses	2,235,710
Net From Operations	(262,067)

Appendix E.7

February 3, 2002

Hi Dave.

Laboratory Animal Services is a unit within the College of Veterinary Medicine that provides specialist services to faculty members who use animals for teaching or research.

The Unit maintains several different animal facilities. Each facility is dedicated to a species of animal and varies in terms of sophistication of facilities and resources. For example, horses may be kept on pasture with run-in sheds, in stalls, or in high security stalls for work with infectious diseases. LAS also has facilities for veterinary care of these animals and even surgical facilities. LAS provides housing, care and veterinary services to several species including mice, rats, guinea pigs, rabbits, cats, dogs, sheep, cattle, horses, pigs, fish, birds (including poultry) and more.

The nature of the services ranges from basic food and care, routine observation for general health, consultations with veterinarians specializing in laboratory animal care, etc. In some cases, LAS provides all of the veterinary care of animals. (In others, where the researcher or teacher is a veterinarian, that person may elect to provide veterinary care him or her self.) The Unit finds itself responsible for advising investigators and teachers on the relevant local, state and federal regulations, and also responsible for making sure that those regulations are not broached. (This is sometimes quite a difficult position for them.) So, they will help with writing of animal care and use protocols, and then scrutinize those to make sure they comply with regulations and are passed by the Institutional Animal Care and Use Committee (a federally-mandated committee).

Once the paper work is in order, LAS will order or procure animals from vendors, arrange for transport, acclimate animals to new surroundings, house feed and care for the animals. The also provide some technical support. For example, relatively few research technicians know how to hold and handle mice or rats so that neither they nor the animals get hurt, etc... After the research or teaching is over, LAS will help with adoption, sale, etc, of the animals. (Some stay for ever, like the R-barn cows.)

I hope this helps you understand. You may have to condense it for your project.

In a formal sense, I would say their mission is to:

Provide physical and human resources for the excellent care and health maintenance of animals used in teaching and research at the College of Veterinary Medicine.

Let me know what else you need.

Dr. Gilbert

Appendix E.8

Summary of a Telephone conversation with Dr. Gilbert on Monday March 3, 2002. (responses by Dr. Gilbert are in blue, the points that I based my questions around are in black and numbered. Also the responses by Doctor Gilbert are not always in full exactly how he stated them, but rather formed from points that I made as the conversation progressed.

The main focus of this interview was to obtain a slightly better idea of the non-functional requirements.)

1) time/space bounds: workloads, response time, available storage space, etc.
for example, if all of these remote LAS locations want to submit reports into the central server, how many can submit at the same time? (i.e. how many labs are there that can possibly want to submit simultaneously?)

Dave, this would be limited by the number of employees, and the distances between sites. Practically speaking, I can not really imagine that two LAS sites would want to submit information simultaneously. Even if they did, a short wait would be permissible. It may happen that accounting or management want to access the system simultaneously, and simultaneously with someone wanting to submit data.

2)reliability
must the system be running 24/7, or are there times when nobody would want access to the information either at the centralized source or at a field location? (i.e. should the system be accessible from places around the clock?)
are there times of the month when necessary maintenance can be performed. perhaps backup of data made to minimize actual disk space needed (perhaps backed up onto a writeable cd?).

The system needs to run 7 days a week. It can be shut down for maintenance, provided that is scheduled. We shut most of our systems down from time to time, usually weekends.

3) security
obviously people outside of the LAS system cannot access the system (or not so obviously?). also are there some pieces of info that are only accessible from one end of the data flow (i.e. only accessible by the accounting service, not by the labs themselves?).

Access should be limited to LAS and to accounting. The remote labs probably need very little access – perhaps to census data, so that they can check it. Primarily, they will submit data. Security is quite important because of the sensitivity to animal rights activists. That probably overrides some convenience aspects.

4) operating requirements.

who is operating this system? i assume that you want it so that current staff members will be operating it without to much training. please correct me if i am wrong.

Ideally, current staff members operate the system. A training period (locally, and limited to a few hours per day; would have to be repeated because not everyone could attend the same session) would be fine. Some sort of manual would be ideal to allow some training of new recruits without a system specialist.

also, do you happen to know exactly what kind of network you are running on right now? that would be useful information since it would contribute to time/space bounds info.

On soft ground here, because I do not know what I am talking about. Here's what they tell me:
NT backbone basically …NT servers, … Microsoft mainly….. IP domain ….Windows 2000….. Make any sense to you?

5) lifecycle requirements
how long do you expect to use this system for?
I
've come to see 5 years as a long lifespan for software. I would hope for 5 years, but not necessarily expect it. Ideally, we will be able to build enhancements or update periodically (every 4 or 5 years?).

b)limits on development.
any time limitations on the development?

I'm not sure I understand fully. A preference: within 9 months. If that is not realistic, well …

lastly:
6) economic requirements:
what are you looking to spend? initially on the development, and then long term?

Like every client I want as much as possible for as little as possible. I have spent about $350,000 on worthless software development, so I am a little punch drunk. For a new system that will cover hospital operations as well as the diagnostic lab, I am looking at about $350,000 in purchase of software and hardware and in personnel expenses to set up, and then about $120,000 in maintenance.
LAS is much smaller. I would hope to come in at $50,000 and $10,000 – is that realistic? Less is good.