Project Plan

SJ SU Library System

October 30, 2001

Joel Frank, Project Manager (Sections III and IV)

Hours Worked:

Signature

Valerie Stanton, Software Engineer (Sections II and V)

Hours Worked:

Signature

Dani Sablan Chalmers, Software Engineer (Section I)

Hours Worked:

Signature
# Table of Contents

I  INTRODUCTION 3

  SCOPE AND PURPOSE 3
  OVERALL SYSTEM SCOPE 3
  PROJECT MOTIVATION AND OBJECTIVES 3
  STRUCTURE OF DOCUMENT 3

II  SOFTWARE ENGINEERING PROCESS MODEL 5

  DEFINITION 5
  TEAM STRUCTURE AND RESPONSIBILITY 7

III  PROJECT TASKS AND SCHEDULING 8

  PROJECT TASKS 8
  SCHEDULING 11

IV  PROJECT RESOURCES AND COST ESTIMATION 12

  PROJECT COST 12
  PROJECT RESOURCES 13
  OVERALL COST BREAKDOWN AND ESTIMATE 14

V  PROJECT MONITORING AND MANAGEMENT METHODS 15

  PROJECT MONITORING METHODS 15
  PROJECT MANAGEMENT METHODS 15

VI  BIBLIOGRAPHY 17

VII  APPENDIX 18
I  Introduction

Scope and Purpose

The library system will support four types of users: students, faculty, library staff, and administrators, each of which will have their own user account and password. The system will support the following online functions for student and faculty users: resource reservation, resource search, account validation, access to current reservation list and check-out list, deadline notification and late fee payment. Library staff will have the following additional capabilities: resource check-in/check-out, record modification, user account modification, and database report generation. Administrative personnel will have access to the following additional database maintenance functionality: top-level user account access and modification, and top-level database modification. The system will provide an interface tailored to each user type.

Overall System Scope

The library system will utilize two servers in order to optimize reliability and minimize downtime. The servers will service clients connected to the local area network. All data including student and faculty accounts, library staff accounts, administrator accounts, and library resources will be stored Microsoft Access Database and accessed through the web. The web interface will be developed using Microsoft Visual Interdev® 6.0 (part of the complete visual studio 6.0). Microsoft Visual Interdev® 6.0 includes support for Active Server Pages and Dynamic HTML, and includes integrated database design and programming tools.

Project Motivation and Objectives

The current library system is outdated and is no longer able to efficiently handle the needs of the students and faculty. Our objective is to develop a superior system that will replace the existing system. This new system will provide a more user-friendly interface and will be able to better accommodate the expanding inventory of library resources. With the latest developments in computer technology and improved processor speeds, the capabilities of this new system will be further enhanced.

Throughout the design process, there are a few main objectives that will be of concern:

♦ Minimize overhead by simplification
♦ Constant communication with customer to keep requirements current
♦ Continual project tracking to limit missed deadlines

Structure of Document

Software Engineering Process Model

The process model section defines the model the team has chosen to apply to our project. Also included are details of the requirements definition, software & system design, implementation and unit testing, integration and system testing, and operation and maintenance. The team structure is included with a diagram to illustrate responsibility.
Project Tasks and Scheduling

The tasks and scheduling section details at great length the list of all the tasks required for the Library System project. Included is the detailed Gantt chart for the scheduling time-line.

Project Resources and Cost Estimation

The resource and cost estimation section details cost estimation using the Line of Code (LOC) method, as well as project resources needed.

Project Monitoring and Management Methods

The project monitoring section breaks down the intended methods for keeping the project development on time and on track. The monitoring section details the weekly team meetings, schedule tracking, meeting with the customer, and process reviews. The management section discussed risk management, requirements management, tracking and oversight, process management, and training.
II  Software Engineering Process Model

Definition

The application development for this project will be done using the Waterfall process model. The steps included in this process are: defining the requirements, designing the system and software, implementing the design and perform unit testing, integrating the design and perform system testing, and finally, overseeing the operation and maintenance (See Figure 1). This is a linear, sequential approach to software development and is often referred to as the “classic” model. We considered some of the other more iterative evolutionary models, but did not find them as applicable to this project. After meeting with the customer to identify overall system needs and requirements, it was evident that the requirements had been clearly defined by the customer and changes to the scope expected to be minimal. In addition, this project is relatively small in size and straightforward in scope; therefore, it was determined that the Waterfall model would be perfectly suited to this project.

Figure 1: Waterfall Process Model
The process steps are defined as follows:

Requirements Definition

In this phase, the customer will be asked to determine the minimum set of features to be supported (the “needs”), as well as the ideal system desired (the “wants”). It is important to establish these priorities as many projects are constrained by time and there are specific needs that must be met within a specific time frame. Once the requirements have been established and prioritized, a more detailed timetable will be determined.

This phase will begin with a careful analysis of the existing process. By examining the existing process, it will improve our ability to fit the design to the way the customer actually operates. The next critical step will be determining how the data is to be organized and what steps will be required to perform specific tasks. In this phase it is important to have the participation of the end users. By addressing usability issues with the users early in the development process, problems may be easier to correct.

Software & System Design

While defining requirements refers to the task of identifying the problem, the design refers to the process of planning the actual solution to that problem. Software and system design is one of the most important steps in the overall development. This process actually consists of two steps. The first is to establish a high-level design that specifies the different parts of the system, and how they interrelate. During this phase, final decisions will be made regarding specific hardware and operating systems requirements, as well as the software tools that will be used. User interface standards will be established to ensure every window has a consistent 'look-and-feel' in its presentation, and is consequently more friendly to the user.

The second part is a detailed specification of each module, window, and function within the system. A detailed database design will be established at this point. Performing a detailed design will allow changes to be made early, while it is still relatively easy. This design specification will serve as a blueprint to guide the development phase.

Implementation and Unit Testing

This phase is mostly self-explanatory, although it usually comprises the majority of the project life cycle. Taking the design established in the previous phase, the application itself will then built and tested at a single system level. This is commonly known as unit testing. Its goal is to make sure that the individual components of the system work well, both separately and in conjunction with other parts of the system.

Integration and System Testing

After the individual components are proven to work well together, the design is integrated system-wide and system tested. The goal of system testing is to make sure that the original requirements have been met, that the requirements embodied in the system are correct, and to verify that the system works as a coherent whole. This stage of testing will be done in close
coordination with the eventual users of the system. A limited-scale deployment of the hardware and software will be done in order to facilitate the testing.

**Operation and Maintenance**

This is where it all comes together. Hardware is installed, and the network configuration is established. The finished application is then installed, and final testing is done to make sure that all of the pieces of the system are working correctly in concert with one another. Any necessary error corrections will be implemented and software corrections will be tested prior to releasing the software corrections to the customer. We will then perform any customer-requested enhancements. As part of ongoing maintenance, we will meet with the customer on a regular basis to discuss upgrades and ensure future compatibility.

**Team Structure and Responsibility**

Our team will be use the controlled decentralized (CD) team structure. The project is relatively simple therefore this model can be successfully applied. The team structure can be seen in Figure 2. Joel Frank will be responsible, as the project manager, for maintaining and tracking the project schedule, supervising the performance of subordinate team members, coordinating efforts between members as well as facilitating all team meetings.

All members of the team have the additional responsibility of tasks denoted in the project Gantt chart with deliverables as stated in said document.

![Figure 2: Team Structure and Responsibilities](image-url)
III  Project Tasks and Scheduling

Project Tasks

The following is a detailed list of the all tasks required for the Library Database Project:

1. Define Project Concept and Purpose
   1.1. Meet with customer and identify overall system needs and requirements
   1.2. Formal Technical Review (FTR) to clearly define project
   1.3. Define Project Scope and Purpose
   1.4. Process Model
      1.4.1. Define Process Model
      1.4.2. Define team structure and responsibility
   1.5. Define project tasks and schedule
   1.6. Assign project resources and estimate cost using the LOC method
   1.7. Define monitoring and management methods
   1.8. FTR to compile project plan documentation
   1.9. Present formal project plan to customer

2. Requirement Definition
   2.1. Define student and faculty requirements
      2.1.1. Resource reservation
      2.1.2. Resource search
         2.1.2.1. Search by Author
         2.1.2.2. Search by Title
         2.1.2.3. Search by Availability
      2.1.3. Account Validation
         2.1.3.1. Password validation
         2.1.3.2. Periodic password change
      2.1.4. Account Access
         2.1.4.1. List current reserved resources
         2.1.4.2. List current checked-out resources
      2.1.5. Deadline notification
         2.1.5.1. Warn user when current date near return date
         2.1.5.2. Alert user when current date is equal to or greater than the return date
      2.1.6. Late fee payment
         2.1.6.1. Pay online using credit card
         2.1.6.2. Pay online via university student account

   2.2. Define additional staff requirements
      2.2.1. Resource check-in/check-out
      2.2.2. Resource record modification
         2.2.2.1. Modify resource title
         2.2.2.2. Modify resource type
         2.2.2.3. Modify resource author
         2.2.2.4. Modify resource storage location
         2.2.2.5. Create new resource record
         2.2.2.6. Delete resource record
      2.2.3. User account modification
         2.2.3.1. Reset user password
2.2.3.2. Create new user account
2.2.3.3. Delete user account
2.2.4. Database report generation
   2.2.4.1. List current resources checked-out by user
   2.2.4.2. List current resources past due date
2.3. Define additional administrative requirements
   2.3.1. Staff/Administrator account modification
      2.3.1.1. Reset Staff/Administrator password
      2.3.1.2. Create Staff/Administrator account
      2.3.1.3. Delete Staff/Administrator account
   2.3.2. Database modification
      2.3.2.1. Create new resource database
      2.3.2.2. Delete resource database
      2.3.2.3. Modify resource database
      2.3.2.4. Create account database
      2.3.2.5. Delete account database
      2.3.2.6. Modify account database
2.4. Define interface requirements
   2.4.1. Student/Faculty interface
   2.4.2. Staff interface
   2.4.3. Administrator interface
2.5. FTR to compile project requirement documentation
2.6. Present formal requirement documentation to customer

3. System and Software Design
   3.1. Design student and faculty modules
      3.1.1. Resource reservation
      3.1.2. Resource search
      3.1.3. Account Validation
      3.1.4. Account Access
      3.1.5. Deadline notification
      3.1.6. Late fee payment
   3.2. Design additional staff modules
      3.2.1. Resource check-in/check-out
      3.2.2. Resource record modification
      3.2.3. User account modification
      3.2.4. Database report generation
   3.3. Design additional administrative modules
      3.3.1. Staff/Administrator account modification
      3.3.2. Database modification
   3.4. Design interface modules
      3.4.1. Student/Faculty interface
      3.4.2. Staff interface
      3.4.3. Administrator interface
   3.5. FTR to compile project design documentation
   3.6. Present formal design documentation to customer

4. Implementation and Module Testing
   4.1. Implement and test student and faculty modules
      4.1.1. Resource reservation
      4.1.2. Resource search
4.1.3. Account Validation
4.1.4. Account Access
4.1.5. Deadline notification
4.1.6. Late fee payment
4.2. Implement and test additional staff modules
   4.2.1. Resource check-in/check-out
   4.2.2. Resource record modification
   4.2.3. User account modification
   4.2.4. Database report generation
4.3. Implement and test additional administrative modules
   4.3.1. Staff/Administrator account modification
   4.3.2. Database modification
4.4. Implement and test interface modules
   4.4.1. Student/Faculty interface
   4.4.2. Staff interface
   4.4.3. Administrator interface
4.5. FTR to compile implementation code and testing result documentation
4.6. Present formal implementation code and testing result documentation to customer

5. Module integration and system testing
5.1. Integrate and test student and faculty modules
   5.1.1. Resource reservation
   5.1.2. Resource search
   5.1.3. Account Validation
   5.1.4. Account Access
   5.1.5. Deadline notification
   5.1.6. Late fee payment
5.2. Integrate and test additional staff modules
   5.2.1. Resource check-in/check-out
   5.2.2. Resource record modification
   5.2.3. User account modification
   5.2.4. Database report generation
5.3. Integrate and test additional administrative modules
   5.3.1. Staff/Administrator account modification
   5.3.2. Database modification
5.4. Integrate and test interface modules
   5.4.1. Student/Faculty interface
   5.4.2. Staff interface
   5.4.3. Administrator interface
5.5. FTR to compile integration and testing result documentation
5.6. Present formal integration and testing result documentation to customer

6. System operation and Maintenance
6.1. Error correction
   6.1.1. Meet with customer to review/compile customer driven bug report
   6.1.2. Compile internally generated bug report
   6.1.3. FTR to review complete bug report
   6.1.4. Implement and test software corrections
   6.1.5. Release software corrections to customer
6.2. Enhancements
   6.2.1. Meet with customer to discuss requested feature enhancements
6.2.2. FTR to review requested feature enhancements
6.2.3. Implement and test feature enhancements
6.2.4. Release feature enhancements to customer

6.3. Adaptation
6.3.1. FTR to analyze market trends to detect possible future incompatibilities
6.3.2. Meet with customer to discuss options regarding software upgrades that will ensure future compatibility
6.3.3. Implement and test required adaptations
6.3.4. Release software upgrades to customer

6.4. Prevention
6.4.1. FTR to review the current software platform and analyze for possible updates that will improve the capability of future adaptation and enhancement
6.4.2. Implement and test desired reengineering tasks

Scheduling

A Gantt Chart detailing the schedule for this project is provided in the Appendix. Shown below are some of the key deliverables associated with each stage of the project.
IV Project Resources and Cost Estimation

**Project Cost**

The project cost estimation was performed using the LOC method and applying it to all major subtasks defined by the project scope.

<table>
<thead>
<tr>
<th>Student and Faculty Functionality</th>
<th>2450</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Reservation</td>
<td>500</td>
</tr>
<tr>
<td>Resource Search</td>
<td>580</td>
</tr>
<tr>
<td>Search by Author</td>
<td></td>
</tr>
<tr>
<td>Search by Title</td>
<td></td>
</tr>
<tr>
<td>Search by Availability</td>
<td></td>
</tr>
<tr>
<td>Account Validation</td>
<td>325</td>
</tr>
<tr>
<td>Password Validation</td>
<td></td>
</tr>
<tr>
<td>Periodic Password Change</td>
<td></td>
</tr>
<tr>
<td>Account Access</td>
<td>400</td>
</tr>
<tr>
<td>List Current Reserved Resources</td>
<td></td>
</tr>
<tr>
<td>List Current Checked-out Resources</td>
<td></td>
</tr>
<tr>
<td>Deadline Notification</td>
<td>225</td>
</tr>
<tr>
<td>Warn User when Current Date near Return Date</td>
<td></td>
</tr>
<tr>
<td>Alert User when Current Date is Equal to or Greater than the Return Date</td>
<td></td>
</tr>
<tr>
<td>Late Fee Payment</td>
<td>420</td>
</tr>
<tr>
<td>Pay Online Using Credit Card</td>
<td></td>
</tr>
<tr>
<td>Pay Online via University Student Account</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Additional Staff Functionality</th>
<th>1550</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Check-in/Check-out</td>
<td>450</td>
</tr>
<tr>
<td>Resource Record Modification</td>
<td>525</td>
</tr>
<tr>
<td>Modify Resource Title</td>
<td></td>
</tr>
<tr>
<td>Modify Resource Type</td>
<td></td>
</tr>
<tr>
<td>Modify Resource Author</td>
<td></td>
</tr>
<tr>
<td>Modify Resource Storage Location</td>
<td></td>
</tr>
<tr>
<td>Create New Resource Record</td>
<td></td>
</tr>
<tr>
<td>Delete Resource Record</td>
<td></td>
</tr>
<tr>
<td>Student/Faculty Account Modification</td>
<td>300</td>
</tr>
<tr>
<td>Reset Student/Faculty Password</td>
<td></td>
</tr>
<tr>
<td>Create New Student/Faculty Account</td>
<td></td>
</tr>
<tr>
<td>Delete Student/Faculty Account</td>
<td></td>
</tr>
<tr>
<td>Database Report Generation</td>
<td>275</td>
</tr>
<tr>
<td>List Current Resources Checked-out by User</td>
<td></td>
</tr>
<tr>
<td>List Current Resources Past Due Date</td>
<td></td>
</tr>
</tbody>
</table>
### Additional Administrative Functionality

<table>
<thead>
<tr>
<th>Function</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staff/Administrator Account Modification</td>
<td>630</td>
</tr>
<tr>
<td>Reset Staff/Administrator Password</td>
<td></td>
</tr>
<tr>
<td>Create Staff/Administrator Account</td>
<td></td>
</tr>
<tr>
<td>Delete Staff/Administrator Account</td>
<td></td>
</tr>
<tr>
<td><strong>Database Modification</strong></td>
<td>770</td>
</tr>
<tr>
<td>Create New Resource Database</td>
<td></td>
</tr>
<tr>
<td>Delete Resource Database</td>
<td></td>
</tr>
<tr>
<td>Modify Resource Database</td>
<td></td>
</tr>
<tr>
<td>Create New Account Database</td>
<td></td>
</tr>
<tr>
<td>Delete Account Database</td>
<td></td>
</tr>
<tr>
<td>Modify Account Database</td>
<td></td>
</tr>
</tbody>
</table>

### Online Interface Definition

<table>
<thead>
<tr>
<th>Interface</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student/Faculty Interface</td>
<td>480</td>
</tr>
<tr>
<td>Staff Interface</td>
<td>400</td>
</tr>
<tr>
<td>Administrator Interface</td>
<td>320</td>
</tr>
</tbody>
</table>

### Estimated Total Lines of Code

<table>
<thead>
<tr>
<th>Code</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6700</td>
</tr>
</tbody>
</table>

These estimations were found by applying the following formula:

\[ S = (S_{opt} + 4S_m + S_{pess}) \]

This LOC estimation was also based on historical data, which showed that the average productivity is 750 LOC.pm. In addition, this estimate is made with the intent of covering all development phases pre and post to the coding phase. Therefore, the labor rate per month is $14,000, and the cost per line of code is $19. Based on this estimate and these current business cost requirements, the LOC estimate for this project is $127,000 with an estimated coding timeline of 9 person-months.

### Project Resources

#### Hardware Resources

Dell PowerEdge 8450 - Server Specifications

- PowerEdge 8450 Pentium III Xeon w/ 2M Cache
- Eight Pentium II Xeon 700Mhz w/ 2M Cache
- External, rack mount tape drive
- Veritas SAN backup software suite
- 32Gb SDram – 32DIMM’s
- OpenManage Subscription
- Dual Raid 32-bit raid controller
- Two 73Gb, 10000-rpm hard drives
- Two Intel Pro 1000 Gigabit NIC’s
- Windows 2000 Advanced Server Operating System with 25 client licenses
- 20/48X CD-Rom
- Electronic CD-ROM documentation
Software Resources

Microsoft's Access database utility will be purchased and implemented into the overall library system design as an off-the-shelf component. This product is fully licensed and validated through Microsoft. The system will also require a web interface tool. Microsoft has a web design studio used for creating web sites/pages and assists developers create dynamic Web sites. Microsoft Visual Interdev® 6.0 (part of the complete visual studio 6.0) includes support for Active Server Pages and Dynamic HTML, and includes integrated database design and programming tools.

Engineering Resources

Engineering resources will be implemented in accordance with the project schedule, which is detailed in the Appendix of this document.

Overall Cost Breakdown and Estimate

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost (in U.S. Dollars)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PowerEdge 8450 Server</strong></td>
<td></td>
</tr>
<tr>
<td>Two are required</td>
<td>$122,681 each</td>
</tr>
<tr>
<td></td>
<td>$245,362</td>
</tr>
<tr>
<td><strong>Microsoft Access w/ Site License</strong></td>
<td>$6,000</td>
</tr>
<tr>
<td><strong>Microsoft Visual Interdev® 6.0</strong></td>
<td>$549</td>
</tr>
<tr>
<td><strong>Engineering Resources (based on LOC model)</strong></td>
<td>$127,000</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>$378,911</td>
</tr>
</tbody>
</table>
V Project Monitoring and Management Methods

Project Monitoring Methods

The project will be tracked using the following methods:

Weekly Team Meetings

Weekly team meetings will be facilitated by the project manager and attended by all team members. Each Team Leader will be required to submit a progress report for each of their respective teams. At this time problems or potential problems will also be discussed. Suggestions will be made for resolution.

Schedule Tracking

The schedule will be regularly reviewed comparing actual milestones to planned milestones as denoted in the project schedule. In addition, the task start dates will be reviewed to ensure the schedule is not slipping. If there are start date delays or if it looks like milestones may not be achieved on schedule, the appropriate Team Leader will be contacted to discuss and resolve the potential delays.

The critical path will be closely monitored to foresee any upcoming project delays. One method for minimizing these delays is maintaining a somewhat flexible schedule and by staffing the team with members who have both similar and complimentary skill sets.

Meet with Customer

There will be regular meetings with the customer to review the progress. Current requirements will be also reviewed to ensure that the project requirements have not changed.

Process Reviews

Monthly software process reviews will be performed to ensure that the project is headed in the intended direction and in a timely fashion.

Project Management Methods

Risk Management

As a part of risk management, sources of risk will be identified, addressed, and mitigated before they threaten successful completion of a project. The two primary risk management elements are risk assessment (identifying, analyzing, and prioritizing) and risk control (management planning, resolution, and monitoring).

Requirements Management

Initial requirements will be established but these requirements are subject to change and must be monitored and managed. Managing requirements will include capturing, tracking, and
controlling requirements, as well as any changes to them. This will establish and maintain a common understanding, between the customer and development team, of the requirements to be addressed by the project. This agreement should be the basis for planning and managing the project.

Tracking and Oversight

The project accomplishments will be tracked and reviewed with respect to the project plan. Corrective action will be taken as necessary based on actual accomplishments and results.

Process Management

In addition to managing the project, the processes involved in managing the project will also be continuously monitored and reviewed. This will include planning, defining, implementing, monitoring, measuring, and improving processes under project management and producing process documentation and improvement plans.

Training

The engineers working on the project will receive any necessary training to support product development. This will ensure individuals responsible for software engineering activities have the appropriate skills and knowledge relevant to procedures, tools, and domain knowledge.
VI Bibliography

Estimates for hardware were determined using Dell Computer’s website, www.dell.com

Estimates for software were determined using Microsoft’s website, www.microsoft.com
Appendix