

Reference Statistics for Olin Library

Feasibility Report



17 February 2006

Ronald DeVera
Matthew Kulick
Yi-Chun Lin
Kavitha Nair
Soutrik Pramanik
Krystal Wang
Kevin Wong

TABLE OF CONTENTS

I.	Executive Summary	1
II.	Preliminary Requirement Analysis	1
	i. Part 1 – Application Overview	1
	ii. Part 2 – Functional Requirements	2
III.	Process to be Followed	4
	i. 1st Iteration	4
	ii. 2nd Iteration	4
	iii. 3rd Iteration	5
IV.	Suggested Deliverables	6
V.	Technical Feasibility	8
VI.	Visibility	11
VII.	Risk Analysis	12
VIII.	Business Considerations	13
IX.	Conclusion	13
X.	Appendix: Gantt Chart	14

REFERENCE STATISTICS FOR OLIN LIBRARY

Feasibility Report

I. EXECUTIVE SUMMARY

The following proposed system is intended for the reference staff of Olin and Uris Libraries. The reference staff clients will be represented by reference specialist Gaby Castro Gessner (primary client) and reference librarian Lynn Thitchener. Ms. Gessner developed the original client proposal and is very knowledgeable about the library's needs and Ms. Thitchener is a project participant. Ms. Gessner will be our primary client. Additionally, Ms. Nancy Skipper will be included in our communications with the client as an ex-officio member. The basic goal of the development team is to implement a computer-based system to replace the Library's current paper records of reference statistics. The new system will allow staff from multiple service points to input data simultaneously and will provide automated reporting of aggregated data to produce hourly, daily, monthly, or yearly reports, cross-tabulated as necessary. The new system will keep the basic functionality of the current paper version, including information on the number of reference questions and their individual durations at various stations in Olin and Uris during a specific period of time. The overall goal of the new system is to streamline and automate this process of gathering data. Successful implementation of the system will enable flexible data output and analysis, which will enhance the ability to make effective management decisions concerning staffing needs, based upon library patron trends and service-need patterns.

II. PRELIMINARY REQUIREMENTS ANALYSIS

Part I – Application Overview

Objectives

The basic functionality of the system will be to collect reference statistics from different service points in the libraries in real time and store the information. Queries can be performed on this data to generate reports based on the statistics for staff allocations.

Business Objectives

The project aims to reduce costs, optimize staff levels, improve customer service, store data more efficiently and in more meaningful categories, and determine "traffic patterns" in Olin and Uris Library. It will keep track of when the reference desks in the library are the busiest. The Library's current paper method will be replaced by a more efficient online system with a user-friendly web interface.

Through this automated data management system, the management hopes to improve their staffing decisions based on workloads at different times of the day and week. The project is also expected to greatly benefit the staff by reducing workload and increasing the ease of data reference-related activities and report generation.

A working prototype will be developed, tested, and implemented in time to be deployed in the next academic semester (Fall 2006).

Current Business Process and Rules

Currently, the client operates a paper spreadsheet at different reference desk locations, using hash-marks to denote individual reference transactions, according to a set of criteria. The criteria are: the time (on an hour range), the date, the nature of the question (technical, reference, directional), the staff level of the assistant (librarians, reference assistants, student assistants), the reference location (Olin Library Reference Desks, Uris Library CL3, etc.), and the medium of the interaction (phone, chat, email, in-person, etc.). In addition, there is an optional comment field for assistants to enter useful keywords or information that they can use when

generating reports or trying to understand patterns of use. The time when the queries are made are recorded on an hourly basis by the reference desk agents. This data is then compiled and entered into a central data repository from which reports are generated. The current system is cumbersome and requires manual tabulation of hash-marks to generate aggregate statistics reports using separate spreadsheet software (Microsoft Excel).

The new system hopes to solve the “hash-mark problem” by eliminating paper based entries, replacing it with a real time system, creating a more efficient centralized central data repository, and generating meaningful reports which can be used for statistical analysis.

User Roles and Responsibilities

Administrator Role: The ability to enter, record, edit, and update data, and exclusive abilities to make limited changes such as adding more users, staff levels, categories, etc.

Other Users' Role: The ability to enter, edit, and store data.

Interactions with Other Systems

The system will be built from scratch using standardized web packages, replacing an existing paper system. Reports generated using this system may be combined with reports being generated by a different system in the library at a future date.

Production Rollout Considerations

The central data repository design and development, the design layouts, and generation of reports are expected to be carried out in a phased manner over three months before the system is tested and put into production. Users are expected to use this system after undergoing a short period of training.

Part II – Functional Requirements

Statement of Functionality

This software system will be flexible and sustainable with a user interface for entering reference statistics into a central data repository.

Users need to be able to view the day's activity specific to a station. The software system will have different access levels so that different types of users may log in and out.

The system needs to allow retroactive editing of input and have the ability to delete older data in case users make entry mistakes.

The system must be accessible from various computer stations simultaneously. At any given time, about 25 users will be accessing the system.

The system needs to enable sorting of data (by name, time, etc). The system should be able to produce a range of reports specific to user needs.

Automatic backups of information from the central data repository should be enabled as well as password-protected user access.

Security and User Capabilities

The software system will support three types of users. In order to access the system all the users will need to login with a password. At the administrative login level, the user will be given additional permissions such as adding or changing staff levels and user types. A limited ability to change categories on the input forms will

also be provided. At the other access level (for Librarians, Student, and Reference assistants), only the data entry and editing functionalities would be provided.

Reporting

The reports generated will help in statistical analysis of the reference data that is collected and stored in the central data repository. Daily, weekly, monthly, quarterly, and yearly reports will be created using the functionalities of the system and sorted data.

Non-functional requirements

The software system will be installed and run on existing Windows systems and the system will be tested out on the library servers. The system needs to be functional whenever reference staff needs to access it.

The criteria for success of the system would be measured by the flexibility and sustainability of the system. The functionality and ability of the system to meet all requirements (i.e. simultaneous access from different workstations, effectiveness of the design of the central data repository, automatic backups, retroactive editing of data, various levels of user access, etc.) would be critical for success as well. Ease of use and efficiency would be adequate measures of performance; after a week of training and testing, the users should be able to use the system effectively.

Optional Features

Ideally, the system should be able to generate data in comma-separated value files, which are known to be easily imported into Excel.

The system may contain a searchable notes field where additional information can be entered via the user interface. Users will be able to search for reference statistics using keywords.

Usability

Usability issues such as speed of operation for the user interface, collection and storage of important quantitative data, speed and efficiency of the work flow processes through automation, and concurrency of collected data will be important considerations.

Scope

The scope of our system includes reference data entry, tabular report generation, and administrative system tasks including user editing, system backup, and limited field editing.

The system will not have general spreadsheet capabilities. It will only perform the calculations necessary to produce the electronic version of the current paper report.

The system will not support critical changes to the fundamental way information is gathered. It will support category renaming and the addition of new user types and locations.

III. PROCESS TO BE FOLLOWED

For this project, the team has decided to follow an iterative refinement approach that involves beginning with a user interface mockup and gradually adding functionality until all of the client's requirements are met. The team chose this particular method due to the client's concern over having a user interface that is easy to use and less cumbersome than the current paper method. The interface is required to be intuitive and simplistic in order to easily collect the reference statistics from the librarians and other assistants. Another version is also required to act as an administrative interface to manipulate the data and perform any necessary administrative functions including report preparation and the modification of data categories. By using an iterative method, the team can quickly prototype a mockup user interface for the client to evaluate. If necessary and time permitting, redesign of the interface will occur based on client feedback. While the interface is being revised at

each step of the iterative process detailed below, the team will be adding in the necessary software modules that will handle the central data repository, report generation, and other functionalities.

Below is the proposed outline of the iteration stages and milestones including what the team expects to have completed at each stage.

Process Outline

User testing throughout: At each milestone, the team will present the most current version of the software to the clients for their testing and evaluation. Although the software will not initially be fully complete during the early presentations, it will give the clients an approximation of the functionality of the final product.

1st iteration (March 10, 2006)

Requirements Document

The team will prepare a formal document that will detail the client's requirements for the software. The client will categorize these requirements into required, desired, and optional features. After the document is approved by the client, the team will design the initial user interface for the client's evaluation.

Mockups: Simple User and Administrator Interfaces

To ensure that the interface designs meet with the client's specifications, the team will develop a user level and an administrative level interface that will have all of the required data entry fields, but will be otherwise non-functional. User level interfaces will contain the proper entry fields to add reference statistics. Initially, these interfaces will be simplistic in nature and may not contain all of the intended fields and functions that the clients propose.

2nd iteration (April 7, 2006)

Design Document and Presentation

A formal document will be prepared that details both the design of the system and the code behind it. In describing the design of the system, the hardware and software needed from the client will be specified. Details about the program design will cover internal functionality so that the client can maintain and modify the end product in the future.

The team will also hold a forty-five minute presentation as a progress report for the client. As part of this progress report, the team will update its task schedules to ensure that the system will be satisfactorily complete by the end of the final iteration.

Revised User and Administrator Interfaces

Based on feedback from the client concerning the first iteration deliverables, the team will modify the design of the user and administrator interfaces as needed. This modification will include more planned features and advanced abilities that will aid in diagnosing users' interaction with the system. Again, during this iteration, the interface may not have all planned features, but will have all those required for this stage of development.

3rd iteration (May 11, 2006)

Final Testing Period

The team plans to reserve two weeks before the final presentation so that the client can test the product in its intended environment with real users and data. All functional requirements will have been met before this point; any changes hereafter will only cover small details, such as aspects of the user interface.

Final Documentation and Presentation

Documentation for the final version of the product will be presented to the client so that the end product can be maintained and extended. This will include information on all required features, which will be fully implemented. The documentation will also cover any desired and optional features that have also been implemented. The team will also provide a demonstration of the system and training so that the client can understand how the user interface is intended to be used.

Final System

The final system will include all features that the team and the client have agreed are required, such as allowing data input and storage, sorting data, and producing reports based on previous data. Based on development progress heretofore, the final system may also include any functionality that the client has also determined as desired and/or optional.

IV. SUGGESTED DELIVERABLES

To satisfy the client's need for a digital and (semi-) automated solution to their current system, the following set of work-products will be delivered to the client:

(1) *Periodic Status Reports*

Throughout the software development process, periodic reports will be written and presented to the client to maintain process visibility and enhance the team's responsiveness to the client's needs. The team understands that the client will wish to comment and respond to the development progress. In turn, the team will strive to continually adjust and target processes and progress to the client's needs. Periodic status reports will detail the feasibility of the project, its exact requirements, its design, and ultimately, its final form and implementation. These will be written documents that are presented to the client and any other individuals the client identifies.

(2) *Periodic Presentations*

Accompanying requirements, design, and final reports will be periodic presentations where the team will demonstrate different aspects of the software system in development. The team's iterative approach to software design means that different presentations may concentrate on specific areas of the client's need. For example, the user interface may dominate particular presentations. These presentations are designed to give the client maximum understanding of how their needs are being addressed in the actual product. Feedback will be critical to these presentations, so that the team may use the client's comments to understand completely how the system meets the client's needs.

(3) *Computerized, Web-based System for Reference Statistics*

This system will be the core deliverable for the client. It will consist of a coded, web-based tool. The client has identified four major functionalities that this system will need to deliver:

- (a) **Reference Data Entry** – This should be equivalent to a hash-mark on the current paper system.
- (b) **Retroactive Editability** – The system will allow certain users to edit features of the reference statistics, and allow delayed statistics entry. In some cases, statistics may be entered after the work day ends. In addition, privileged users should be able to remove or edit entries, and normal users should have a limited ability to delete their own entries.
- (c) **Report Generation** – The system will automatically generate digital reports equivalent in information content and similar in style to those currently being generated. The reports will be generated with varying time granularity (year, month, week, day, etc). Minimal user interaction will be required for report generation.
- (d) **System Backups** – The system will automatically back itself up and allow privileged users to restore backups in case of failure. Additionally, back-doors to the system will be provided for future maintenance.

The deliverable system will consist of a graphical user interface front-end for assistants to utilize the system's functions and a centralized back-end central data repository component where the actual data and program the user interacts with will be stored. Therefore, the system will span web-based interactions and code running on the library server.

(4) *Good Faith Requirements Agreement*

After the project requirements have been discussed and reviewed with the client, a requirements agreement will be presented to the client to clarify exactly what the project intends to accomplish. The agreement will explicitly spell out which features and objectives the team intends to deliver.

(5) *Documentation for Use and Mechanics*

The client will be provided documentation both explaining how to use our system and describing its underlying mechanics. The client has expressed interest in gaining familiarity with the system, and the documentation will be useful for reference needs.

(6) *Demonstration and Client Training*

In addition to documentation, the client has requested training for their staff to use the system. The team will satisfy this need by providing demonstrations of the system (in various stages of completion) throughout the semester (in coordination with our periodic presentations) and by allocating time after the final system is finished to train the clients in the use of our system. The demonstration will consist of performing routine tasks that have been identified by the client, and the training will either consist of group instruction led by team members or one-on-one training with the client.

V. TECHNICAL FEASIBILITY

The feasibility of the technical requirements can be judged by identifying and outlining *at least one* technical method that will satisfy the client's needs. Since any technical solution must embody the client's requirements, it is useful to identify them:

Requirements:

- (1) *Data sorting by different fields (time, date, etc.)*
Each digital "hash-mark" will need to include different fields of information since it will be desirable to be able to sort and organize the data in different ways depending on the user's preference.
- (2) *Centralized data repository*
Since many locations will be operating at once and more locations may be added, a centralized data repository is needed to coordinate all of the information. Since different locations may be active at different times, stations should be completely independent of the functioning of the total system.
- (3) *Multiple levels of access to the system*
Users include librarians, reference assistants, and student assistants, and the allocation of staff is one of the important goals of the system. Therefore, different levels of staff need to be identified by the system, instead of using separate data sheets.
- (4) *Retroactive editing of input data*
Data may need to be inputted much later, especially if the library is busy, or it will need to be corrected or deleted.
- (5) *Multiple, simultaneous users and input*
Many reference assistants may be working simultaneously and everyone will need a way to enter their data without waiting for someone else to finish on another computer.
- (6) *Multiple, remote access points*
The reference locations will be spread across multiple libraries and desks. All these locations must be able to access and use the system.
- (7) *Administrative interface*
In the future, an administrator will be needed to add new access levels, change categorical descriptions, or make certain modifications to the central data repository or reporting system.
- (8) *Automatic report generation*
The new system will generate reports as one of the main goals of switching to an automated, computerized system.
- (9) *Security*
Security needs to be maintained to ensure the integrity of the data, although no malicious or advantageous editing is expected. Password-protection and a login system (based on access-level or user-type) are sufficient.

Given this rough outline of requirements, a possible system design could be the system in Figure 1.:

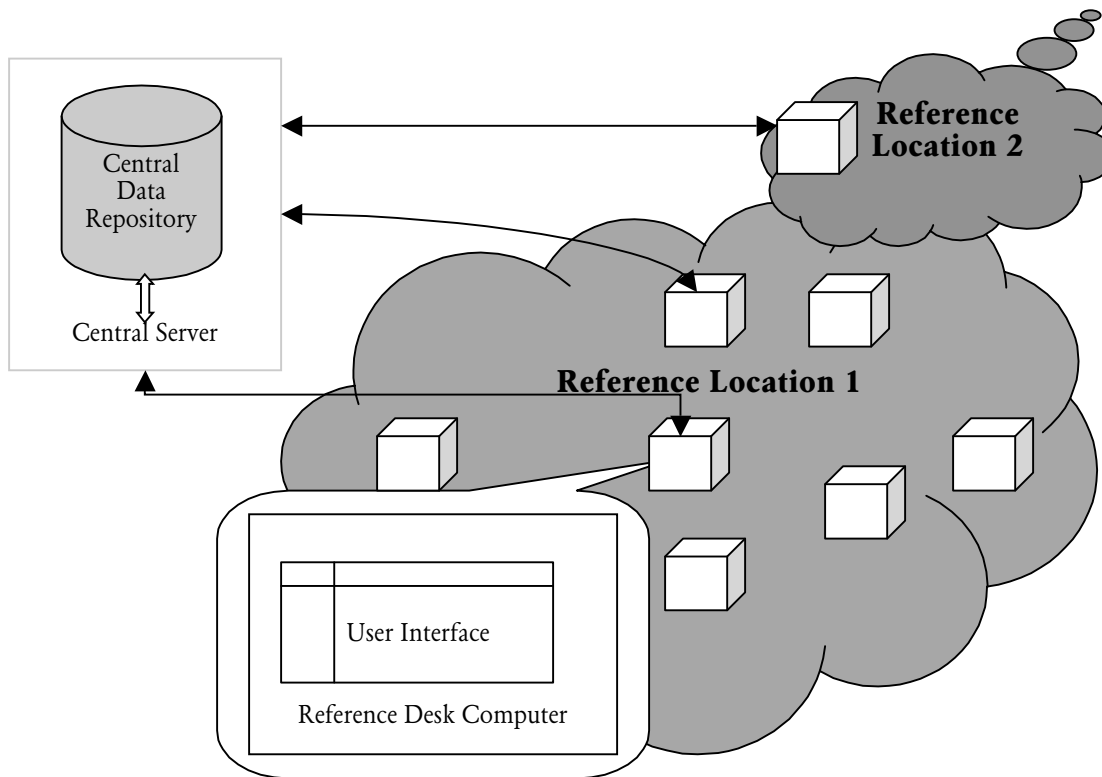


Figure 1. An abstract representation of the system design. Multiple reference desk computers will connect to a single central data repository. The cubes represent individual reference desk computers.

In the above figure, a central server contains a central data repository of all of the reference statistics that need to be tracked. It also contains information about the potential types of users and the reference locations, categories, etc. The central server itself contains backups of the central data repository.

The different reference locations, represented by shaded clouds, are places such as Olin Library’s Reference Desk. Each reference location’s computers use web-based graphical user interfaces to connect remotely to the central server and access the central data repository. Each computer can access the central server and central data repository independently, represented by the black arrows, which are network/internet links. In actuality, this means that the computers (cubes) in reference location 1 are just as independent from each other as they each are from any other computer in reference location 2.

To achieve this setup, the system will require a central data repository, server-side scripting to interact with this repository, and web server software to host the entire system. One possible system configuration would use the most recent versions of MySQL, PHP, and Apache HTTP Server:

- The central data repository can be implemented with a MySQL database, which would support data storage and editing.
- The server-side scripting can be implemented with PHP, which would support different access levels, editing and retrieving database information, report generation, and user and administrator interfaces.
- The web server can be run with Apache, which would support simultaneous users and access points.

In conclusion, **there is at least one technically feasible solution to the proposed system.** This feasible system would consist of a centralized MySQL database, an Apache HTTP (Web) server, and a PHP-coded web interface. The combination of these freely available software products and the team’s own coding will satisfy the client’s requirements.

To further test the feasibility of this possible system, the team must consider that the client expects roughly 20-30 users to be able to simultaneously access the system. With the limited number of end-users, the hardware limitations of the library’s central server should not be a problem.

Finally, it should be noted that the final system delivered to the client may be **different from the technically feasible one described herein**. The purpose of this exercise was to determine if the project itself was feasible *at all*. Future concentration on the requirements of the system will be made and an optimal architecture will be adopted. A preliminary Gantt chart has been attached an appendix for an approximation of our development process and task schedule.

VI. VISIBILITY

The team will take efforts to maximize the visibility of the system and the development process. This will ensure that the project is being developed in line with client specifications. Any deviations from those specifications can also be caught early and corrected through client feedback. Various visibility methods the team intends to use are described below.

Communication

In person meetings and emails would be the primary form of open communication to keep the clients updated with the progress of the project. Regular meetings will be held with the client to discuss progress and for the purposes of two way feedback. The team will also meet as a whole at least once a week to assure all members are caught up and understand their roles and jobs.

Intermediate Deliverables and Presentations

Live demonstrations: The client will be given demonstrations of the progress through presentations at the client site and at the monthly presentations corresponding to each major phase in the project.

Presentations: Slideshows of design layouts of screens, reports and demos of working functions, and the system will be shown to the client to keep them updated with the team's progress.

Reports: The clients will also be presented with copies of the documentation, which record details at each phase in the software development process. These progress reports will also enable them to be well aware of the details of the project from their perspectives.

VII. RISK ANALYSIS

As with any project, this undertaking is not entirely risk-free. Three major risk categories have been identified: time, resource, and functionality.

Time Risks

As course requirements specify that the project must be completed within one academic semester, any extensions are not possible. This introduces the risk that the system may not be completed with the full functionality the client wants within the given timeframe of a semester. In this case, there is also a second risk of delayed implementation of the system if the client chooses to wait until the system is fully functional.

Resource Risks

Resource risks involve technologies the team has available for their use. Due to costs and other external constraints, the team may not be able to obtain the needed or best resources to complete parts of the system. For example, one identified resource is that team members will need laptops running a Windows operating system. Luckily, Cornell's Computer Science department has supplied this resource to those team members in need of laptops.

In addition, there are inherent risks in the resources, such as the software and hardware the team decides to use. Currently, to keep costs at a minimum, the team is considering open-source software, which is available without charge. Hardware for actual system implementation is also dependent on the hardware the library has available. Risks that stem from resources include hardware failures, system crashes, bugs in the code, etc., which may cause accidental data loss. Changes in the computer system of the library in the future that largely cannot be foreseen may also cause the system to malfunction. Since part of the system is web-based, slight variations in display of the user interface may also occur due to different internet browsers.

Functionality Risks

Functionality risks have to do with how the system works. Issues that fall under this category include developing a user interface that is not user-friendly or not well-liked by the client, or producing functions that have limited sustainability. The biggest risk comes from developing a system that does not do what the client wants it to do.

Out of the three risk categories, functionality risks are the easiest to reduce since functionality constraints are more flexible than time or resource constraints. However, minimizing functionality risk is usually accomplished by omitting specific parts and/or functions of the system, as decreasing functionality naturally decreases its associated risks. The team would like to avoid doing this as much as possible. The clients must be aware that it is possible that this must be done in order to deliver the system by the due date at the end of the semester.

Risk Management/Minimization

Having outlined the basic risks associated with this project above, the team is prepared to take precautionary actions to minimize these risks. The principle plan is to develop and practice good management strategies. The team intends to divide the project into a series of iterative phases that have concrete milestones as discussed in previous sections. These milestones will provide project visibility and allow the client to see the team's progress at each stage. Multiple mockups will allow the client to evaluate usability, increase familiarity with the system, and shorten the learning curve.

Frequent communication and feedback from the client are also essential for client satisfaction with the user interface and functionality. The team will also constantly review their progress and modify goals if necessary to deliver a satisfactory system on time to the client.

VIII. BUSINESS CONSIDERATIONS

There are several business considerations that must be taken into account when determining the feasibility of the Olin Library project, including but not limited to: disclosure of trade secrets and sensitive information, copyright and trademark issues, and considerations with regards to patents.

Trade Secrets and Sensitive Information

As far as the team could gather from discussions with representatives from the Olin Library, there are no trade secrets or sensitive information that will be dealt with in the implementation of our system. Since much of the information deals with user question types and usage numbers in the Cornell campus libraries, the information is not of a sensitive nature that will have to be heavily guarded through several security measures. Even in light of this situation, the system will be designed with password protected pages to prevent malicious users from accessing and corrupting this valuable data.

Copyrights and Trademark

Since this project is being completed for the Olin librarian team, the team intends to give Olin library a limited license to use and modify the system. A preliminary agreement is as follows:

The CS 501 team consists of the following members: Ronald DeVera, Matthew Kulick, Yi-Chun Lin, Kavitha Nair, Soutrik Pramanik, Krystal Wang, and Kevin Wong, hereafter referred to as “the team.” The team will give a limited license to Cornell University, Olin Library librarians, and related staff, hereafter referred to as “the client”. This limited license will allow the client to use and modify the software system for an unlimited period of time. The team will not be responsible for any modifications after the software system is delivered, but will help with any questions or concerns of the client as time and circumstances permit. The team reserves the right to be able to demo the software system to prospective employers and showcase the software system as a work created by each team member.

Since the team does not plan to trademark any names in relation to the software system, trademark are not foreseen as being an issue.

Patents

No part of the system is foreseen to be eligible for any patent applications. However, if upon a later date, a part of the system is found to be patentable, the team reserves the rights to the uncontested patent and any derivative works based therein, while the client will automatically gain non-exclusive rights to use the system, and will have full rights to the use and modification of the system regardless of any patent rights held by the team.

IX. CONCLUSION

From the results of the feasibility study, the team finds that the Reference Statistics for Olin Library project is feasible in terms of technicality, skill of team members, and time. Given the time constraint of one semester, the team believes the scope of the project is manageable and that the client’s requirements can be satisfactorily fulfilled upon system completion. The team members also possess the adequate skills to implement the system and are familiar with hardware and software that may be used in this project. The conclusion of the feasibility report is to go ahead with this software development project.

X. APPENDIX

