CSE2101 - SOFTWARE ENGINEERING 1

Software Project Plan

Laboratory Information Management System (LIMS)

Prepared For: Sigma Laboratories

Prepared By:

Kenard Isaacs

USI#: 1040368

TABLE OF CONTENTS

INTRODUCTION

This document serves as the project plan for the Laboratory Information Management System (LIMS) software development effort for ‘Sigma Labs & Medical Supplies’. Sigma Labs has been operating using a paper-based system for records. This project has been initiated to replace the current form of record keeping method with an information system that is integrated into up-to-date data interoperability efforts, efficient and secure in order to better aid Sigma Labs’ mission.

The paper-based recording process is inept and tedious when it comes to receiving, tracking, reporting test results, and billing. This has greatly impacted the amount of time Sigma Labs has to service customers during the day and has, seemingly, caused prolonged workflow. The LIMS project will implement the ability to report testing results and easily share information between locations via electronic data transfer. The LIMS project will also enjoy the additional benefit of adding functionality based on modernized technology. This expected effort is to reduce the maintenance requirement for IT resources. (Sigma Labs has opted for this software company to handle maintenance of the system after implementation).

Objectives

This project plan will contain all of the information about the development plan and the development cycle. It will designate each member roles and responsibilities in terms of development as well as how and what methods they will utilize to accomplish their tasks. It will also set timelines to ensure that the project is completed on schedule. It will adhere to the standards stated in the most recent version of the SRS signed by the customer. The UI design will be included in the initial prototype. It will be able to show client prompted login, browse between different pages, contain the client's logo, and the remaining texts.

Constraints

Budget and Phases

The current budget of the LIMS system is $20, 000, 000.00 GYD and is planned to be implemented August, 2023. The project will be managed in five (5) phases along with several funding gates within each phase:

Phase One (1): Requirements Definition

Funding Gate 1 - 2

- Select and commission an Environmental Scan Committee.

  Environmental Scan:

  - Determine the relevant information required to build the LIMS, constraints and set goals.
  - Cost Benefit Analysis (CBA): compare costs between solutions and determine the most efficient strategy for building the system.

- Select and commission a project management Quality Assurance Consultant on behalf of the firm (if requested by Sigma Labs).
- A detailed project management plan along with a Software Requirements Specification (SRS) document for the LIMS software will be developed.

Funding Gate 3

- Accept delivery, review and assess the Environmental Scan.

Phase Two (2): System and Software Design

Funding Gates 4 - 5

- The allocation of the required hardware and software necessary to build the system.
  - Hardware - necessary computers, network devices, etc.
  - Software – cloud storage (AWS), licenses for technologies (e.g., Oracle RDBMS).

- The building and design of the system.
  - The coding and design of the system.
  - Periodical prototyping

Phase Three (3): Implementation and Unit Testing

- Rigorous and repeated testing components/modules of the software.
- Make improvements if necessary.
- Rinse and repeat.

Phase Four (4): Integration and System Testing

- All components/modules of the software are combined.
- Software is tested as a whole to check if requirements are met.
- The software is tested with dummy data.

Funding Gates 6 – 7 (- 8)

- Hardware and network setup and distribution among branches is implemented.
- Software in integrated into business.
- Assimilate all records into system.
- Retest.
- Train staff to use LIMS.
- System's efficiency: Subject Matter Experts (SMEs), if requested, for each testing group will be brought on to reflect internal and external stakeholder requirements. Functionality, convenience and effectiveness will be monitored at each survey stage. SMEs will provide feedback regarding their User Experience (UX).

Phase Five (5): Operation and Maintenance

Funding Gate 9

- Anticipated maintenance cost will be $1, 500, 000.00 GYD per year for a ten (10) year period prior to implementation.

The project will commence on the 26th September, 2022 and will conclude on the 31st August, 2023.

Specialized Software Constraint

The software under development is intended to be specialized software that is strongly focused on the demands of the customer. As a result, the program is not intended to service a large market or various clients, and its implementation is not needed to be compatible with different platforms or interface standards other than those specified in the application design restrictions.

Constraints and Assumptions for Developers

1. Team members will attend all meetings.
2. Team members will meet all the deadlines.
3. Team members will follow the requirements specified in SRS.
4. Team members will not be allowed to access the system after implementation.

Limitations

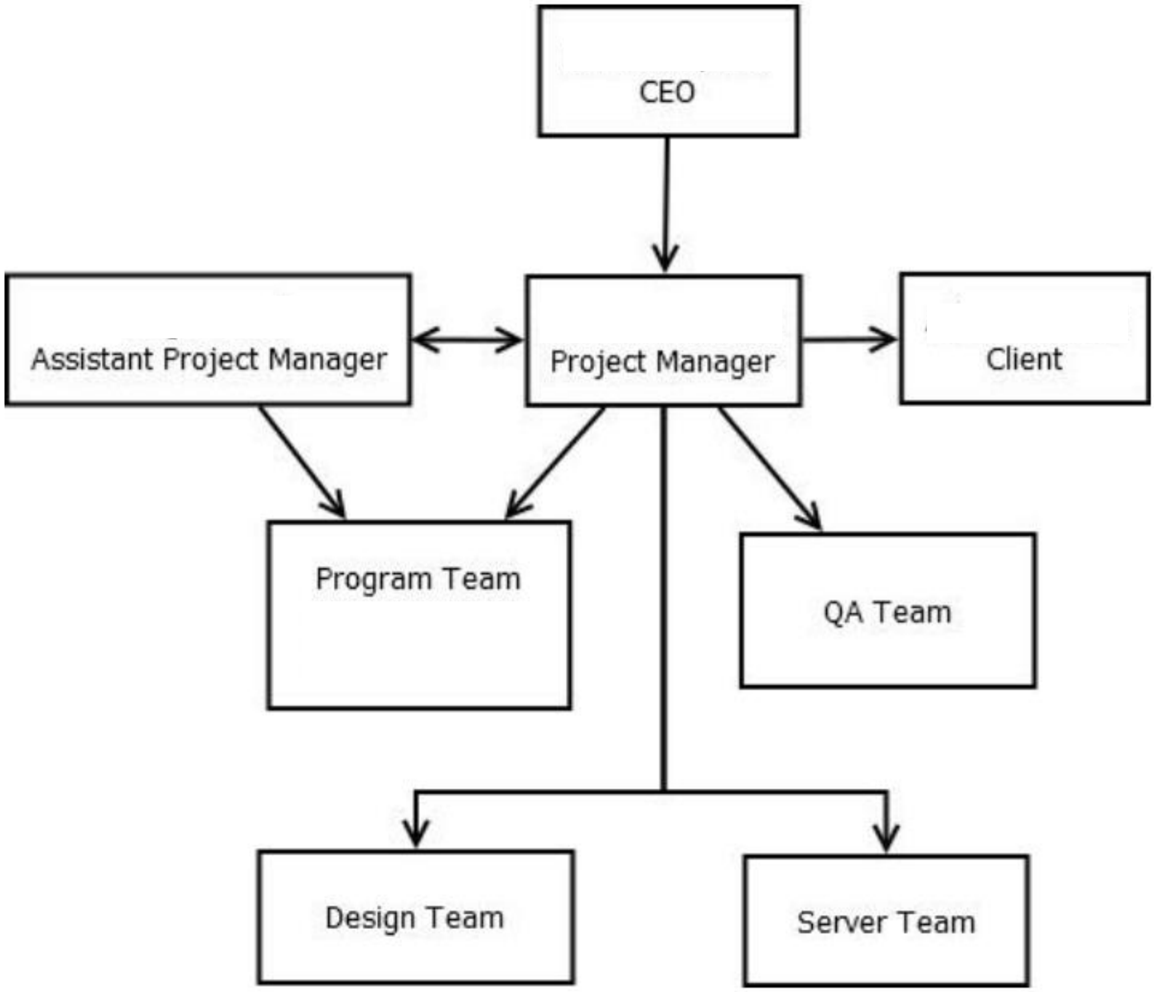
To explain requirements and/or design components, the software development team may create mockups and prototypes of displays, reports, and procedures from time to time. Although some of the prototypes appear to be rather substantial, they are more akin to a movie set in that everything appears to be good from the front but there is nothing in the back.

When creating the prototype, the developer writes the bare minimum of code necessary to describe the requirements or design components under consideration. There is no attempt to adhere to code standards, offer effective error handling, or interact with database tables or components. As a result, retrofitting a prototype with the essential features to build a production component is often more expensive than developing the component from scratch using the final system design document.

For these reasons, prototypes are never designed for commercial usage and are often handicapped in some way to prevent end-users from mistaking them for production components.

PROJECT ORGANIZATION

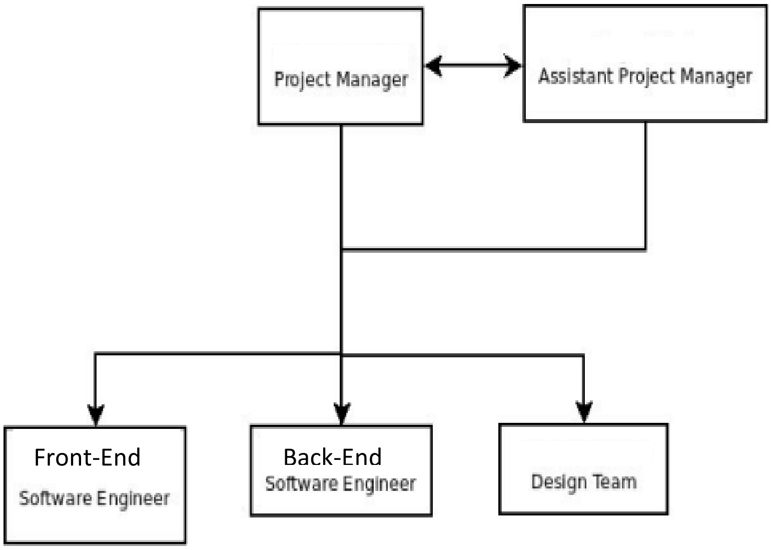
External Interfaces



- CEO- Monitors performance of the development team and provides guidance to the project managers.
- Project Manager - Fosters communication between the client and the team. Uses communication resources such as email and basecamp to communicate between the different groups associated to the development team.
- Assistant Project Manager - Monitors team progress in accomplishing development goals and manages documentation resources. In the case of Project Manager absence, can take over duties and responsibilities associated with the Project Manager.

- Client - Communicates to Project manager the requirements of the LIMS and gives feedback on presented prototypes.
- Program Team - Codes the required objects and combines the illustrations from the Design Team into the working prototypes.
- Design Team - Illustrates the required graphics used in the LIMS.
- Server Team - Process requests for server support relating to the LIMS.
- Quality Analysis (QA) Team - Evaluates prototypes as they are made available and tests for worst case scenarios.

Internal Structure


Project Manager
Assistant Project Manager
Front-End Software Engineer
Back-End Software Engineer
Design Team

Roles and Responsibilities

- The project manager and assistant project manager work together to facilitate progress in completing the prototypes.
- Project Manager - Communicates to the team what the client desires in the LIMS software.
- Assistant Project Manager - Works with project manager to complete prototype development.
- Software Engineers (front-end) - Builds UI components and Implements designs from Design Team.
- Software Engineers (back-end) - Builds API call and makes requests to the database.
- Design Team - Provides Illustrations and design consulting to the development teams.

RISK ANALYSIS

For the purpose of this article risk is defined as the chance or probability of an event occurring that may alter the progress or outcome of the LIMS. Risk Analysis is defined as the identification and assessment of the risk, and the levels of each risk, faced by the project. In addiction to the identification and assessment of risks are approaches that will be taken to reduce or management risk at various stages of the systems' development life cycle.

There are three (3) significant areas of risk within the SDLC, namely: project definition and start-up, evaluation and development of the system, and implementation of the system. Within these stages are different risk factors. For simplicity reasons and stature of the project, only five (5) of the most significant risks associated with the LIMS (throughout the SDLC) are highlighted. These risks are shown in the table below with only a best-case, worst-case scenario or outcome (Low Risk, High Risk). The team leader did not want to measure the probability of these risks happening but did this to cater for the worst possible circumstance (provide no probable or improbable grey area). Anything less than the low-risk circumstance is treated as a possible high risk.

RISK FACTORS	**CIRCUMSTANCE**	
	LOW RISK	HIGH RISK
Personnel providing application knowledge, project members who have experience of the business area.	Know about IS and user areas, all.	Lack user area knowledge, none.
The new system	-	A completely new system.
Technology mix (database, network, etc.).	-	New and complex architecture.
Sympathetic users identified for prototyping, testing and implementation.	Yes.	No.
Performance of development system matches expectation.	Yes.	No.

1. *Knowledge of the project members and users*

The skills of all of the project team members should be assessed before the start of the project. Clearly determine if the software development team members have been involved successfully with similar projects in the past, especially within the same area, there will be high degree of confidence and technical skill. In contrast, a team which is new and has little experience will require team building and technical training, ideally before the project starts. The likelihood of this risk factor arising would be due to a poor vetting process of project management of both the project team and organization staff that will be using the system.

Equally the experience of the user representatives working with the project team members may not be adequate enough to get a high-performance team working immediately. Therefore, some on-the-job training in project team membership may be appropriate.

An issue of major concern will be the degree and depth of understanding and knowledge each member has in the other team member's disciplines. When there is none, a level of common understanding has to be developed. Equally important is the degree of knowledge and understanding the computer staff have in the laboratory. A corollary is that carefully trained computer staff must be retained by the project, otherwise momentum will be lost. Here team management, has the responsibility to ensure that the team is formed, and has the required level of knowledge and understanding to do the job, before the project is fully underway to reduce risk.

2. *New system*

A new system tends to impact on many areas, staff might be unsure of their duties and responsibilities when it becomes operational or they could resist its introduction. This risk factor is how the highest priority. To counter this, the team will try to assess the impacts of the new system on the organization and the users. Communication of the benefits of the new system to the users should be undertaken but remember to keep the statements realistic to manage expectations of the users. This discussion can be achieved in groups or individually in person.

Involvement of the users in all stages of the project is essential. Areas where this can occur are: planning the project, analysis, testing the prototypes and implementation. Request input on how to structure and phase the training to use the new system. A "champion(s)" for the system will be identified and involved throughout the project from the organization.

3. *Technology mix*

The greater the number of technologies that have to be integrated into a system environment the greater the risk becomes. A number of new technologies that seems unwieldy to untrained or inexperienced staff of the organization can cause resistance to the new system. Also too many technologies can result in misuse or possible malfunction, which can lead to system failure. Wherever possible, the team will try to keep to the simplest approach consistent with the requirement of the system and meet the needs of the user and organization. Wherever possible, the team will use components, or proven technology, that the organization has knowledge of (and has possibly used successfully before).

At this point in planning the LIMS project team is focusing on keeping a desktop workstation as the only focal point for the system. The team will try to make the UI/UX design as user friendly as possible. However, to make this system as efficient as possible, the team will stack together numerous of the latest and most innovative technologies available to develop the software. These will be used to build the insides if the system and hence, the user will not be interacting with them directly.

4. *Involvement of users in prototyping and testing*

Before development starts, the project team will identify a group of 'sympathetic' users, who will be used to test prototypes or functions developed via conventional programming. The users will represent all groups within the laboratory environment. Credibility is easily lost during development by word of mouth and the actual performance of the system. There is little point in selecting a group of users who do not want the system to succeed or are skeptical to the use of automation. What is required is constructive comment and criticism that will allow detrimental comments about the system being made to everyone.

5. *Poor system performance*

This is the classic reason for failure of projects during implementation: the system was sized either by estimation or by a formula. The overall system performance is not sufficient to operate the system effectively and provide adequate performance to the users and ultimately the laboratory customers. Effectively, the system is useless an unable to perform its function. This can be due to a combination of factors:

- Hardware related: processor undersized, insufficient memory, etc.
- Software related: inefficient or non-optimized software routines, database searches slow and not optimized; estimates of laboratory workloads too slow.

A number of ways can be used to overcome this problem. One is to define the overall workload of the laboratory accurately together with defining in unambiguous terms what a sample, test, analysis and result means within the context of a specific laboratory. This will allow development team to size the system more accurately. Another is to carry out performance test on the potential system and configuration and time the responses obtained. Finally, specify the minimum response times required and try to improve system before integration.

HARDWARE AND SOFTWARE RESOURCE REQUIREMENTS

The hardware and software requirements of the LIMS will not be divided into section, based hardware and software components, but will be grouped according to the systems configuration for installation.

Workstation Requirements

LIMS client applications are user-friendly Graphical User Interface (GUI) with the majority of functionality available via the web. The client/user (Desktop) workstation should have the requirements listed below. LIMS is a stand-alone product, and does not require any additional software or utilities.

Workstation requirements for single-user installation or each workstation in a multi-user installation:

1. 2GHz or faster processor.
2. Minimum of 32GB RAM.
3. 1TB disk space.
4. 1920 x 1080 resolution monitor.
5. Windows 10 professional, Windows 11 professional.
6. Microsoft Office.
7. Available USB port if barcode hardware will be used.
8. Connection to local or network printer for report generation.

Additional workstation requirements in a multi-user installation:

1. Cabling
2. Login account for users
3. Read, write, create and delete file privileges in the system's BD.

Server requirements

LIMS uses Oracle RDBMS and middleware software which is run on an Amazon Web Services (AWS) server. The high reliability if this Oracle database, its ability to handle huge amounts of data, its integrity, security and portability have made it the best choice for the system. A stack of web technologies will also be used to used to build the software's UI/UX design around the Oracle database (since Oracle APEX does not provide much functionality), the necessary API's and the rest of the software. Fee's will have to be paid to be able to use AWS and Oracle services.

Network Requirements

LIMS is a true client-server application. The server application is separate from the client application. Design of the logic and business rules keeps network traffic to a minimum. Special considerations allow the client application to reduce the volume of transmitted data through the network. This is especially useful for slow and remote connections.

LIMS can be deployed across internet protocol (IP) connections using Citrix or Microsoft terminal server mode, to allow users straightforward access throughout the organization, at remote or home offices. The network will require a bandwidth of 200Mbps or faster from a reliable Internet Service provider (ISP) and Ethernet connections from router to all workstations, switches and routers.

WORK BREAKDOWN

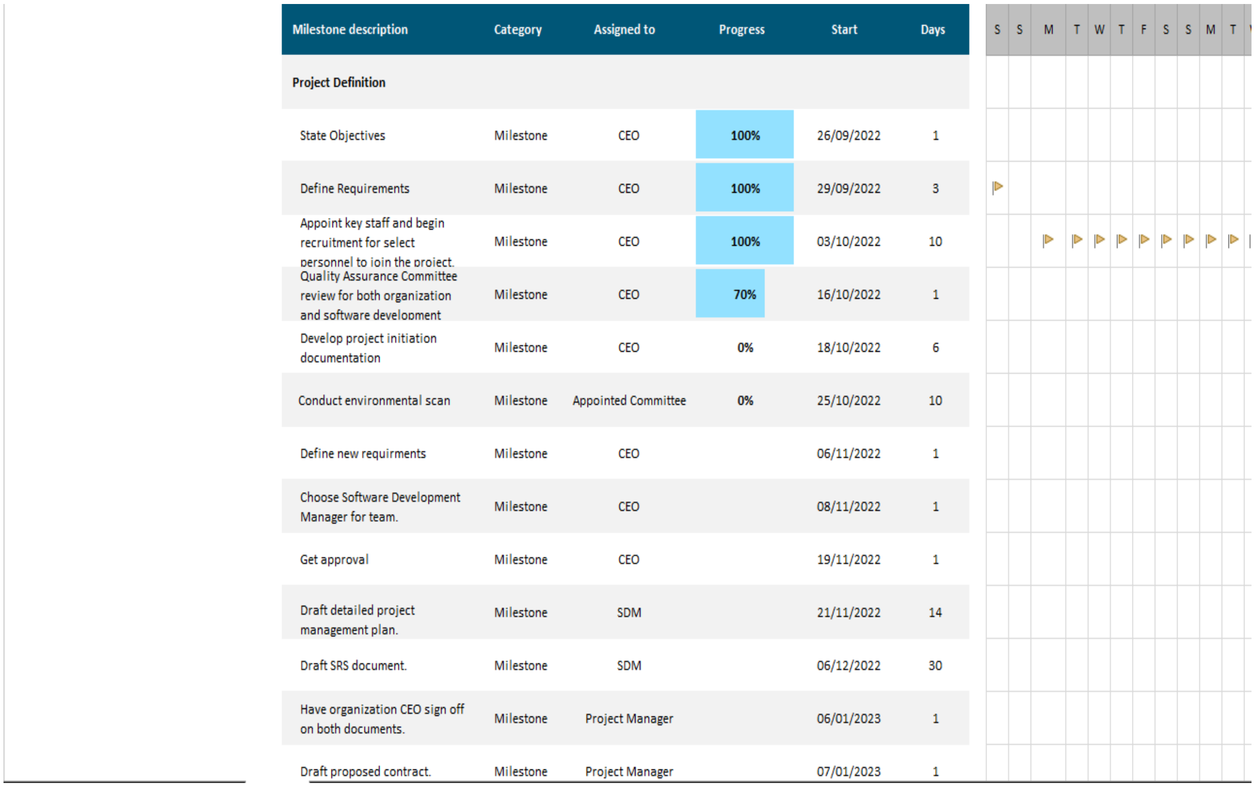
The work breakdown for this LIMS project is presented in the following table.

ACTIVITIES	MILESTONES	DELIVERABLES
Project Definition	State objectives • Clarify request. • Establish objectives • Identify key issues	Detailed Project Plan, SRS document and contract for system project.
	Define requirements • Obtain current documentation	
	Appoint key staff and begin recruitment for select personnel to join the project, including business resources needed to complete high-level requirement elicitation and document.	
	Quality Assurance Committee review for both organization and software development team.	
	• Develop project initiation documentation. • Post QA RFP to solicit for developer team, evaluate proposals, conduct interviews.	
	Conduct Environmental Scan	
	Define new requirements	
	Choose Software Development Manager for team.	
	Get approval	
	Draft detailed project management plan. Draft SRS document. Have organization CEO sign off on both documents.	
	Draft proposed contract.	
	Sign proposed contract and initiate project.	
Design	Functional • Identify interfaces • Design I/O	Prototypes of the UI/UX

	• Spec audits/controls • Confirm specs	
	Technical • Define program specs • Prepare system flow • Convert data • Build integration test plan	
Implementation	Programming • Source code ○ Construct code ○ Conduct unit test • Conduct system test • Documentation • Get approval	Prototypes, functionality of successfully tested modules, integrated system with test data, finished product.
	Installation • Testing ○ Finalize test plan ○ Create test data ○ Conduct test • Training ○ Conduct operations training ○ Conduct user training • Cut-over ○ Finalize plan ○ Convert data ○ Cut-over to production • Get approval	
	Operation • Operate system • Review ○ Establish plan ○ Review performance • Audit	

PROJECT SCHEDULE

The Gantt Chart shows the dependencies between activities, the estimated time required to reach each milestone, and the allocation of people to activities.



Milestone description	Category	Assigned to	Progress	Start	Days	S	S	M	T	W	T	F	S	S	M	T
Project Definition																
State Objectives	Milestone	CEO	100%	26/09/2022	1											
Define Requirements	Milestone	CEO	100%	29/09/2022	3	▶										
Appoint key staff and begin recruitment for select personnel to join the project.	Milestone	CEO	100%	03/10/2022	10			▶	▶	▶	▶	▶	▶	▶	▶	▶
Quality Assurance Committee review for both organization and software development	Milestone	CEO	70%	16/10/2022	1											
Develop project initiation documentation	Milestone	CEO	0%	18/10/2022	6											
Conduct environmental scan	Milestone	Appointed Committee	0%	25/10/2022	10											
Define new requirments	Milestone	CEO		06/11/2022	1											
Choose Software Development Manager for team.	Milestone	CEO		08/11/2022	1											
Get approval	Milestone	CEO		19/11/2022	1											
Draft detailed project management plan.	Milestone	SDM		21/11/2022	14											
Draft SRS document.	Milestone	SDM		06/12/2022	30											
Have organization CEO sign off on both documents.	Milestone	Project Manager		06/01/2023	1											
Draft proposed contract.	Milestone	Project Manager		07/01/2023	1											

GROUP 6 MEMBERS

FIRSTNAME	LASTNAME	CONTRIBUTIONS
Kenard	Isaacs	Did the entire assignment
Ashley	Ghanschan	Took leave of absence / no contribution
Denita	Gillis	Took leave of absence / no contribution
Roshawn	Washington	No contribution
Aliyah	Shaw	No contribution
Dwight	Joseph	No contribution