League Management System

Project Statement

Project Manager: Izaak Ford-Dow

January 22, 2025

# Approval Page

League Management Inc.

Leage Management System

LMS

This document serves to confirm the approval of the client, Leage Management Inc., and the project manager, Izaak Ford-Dow, for the outlined project. By signing below, both parties affirm that they have reviewed this document in full and agree to the proposed plan. Acceptance of this plan is confirmed by signing and dating below, allowing the project to move forward as detailed.

|  |  |  |  |
| --- | --- | --- | --- |
| Approver Title | Approver Name | Signature | Date |
| Client | Riley Jamieson (CEO) |  |  |
| Project Manager | Izaak Ford-Dow |  |  |

# Document Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Document Name | Date | Author(s) | Comments |
| 1.0 | CIS2261\_FordDow\_Izaak\_LeagueManagement\_Project\_Statement.docx | 2024/11/25 | Izaak Ford-Dow | First Draft |

Table of Contents

[Approval Page 1](#_Toc188648216)

[Document Tracking 2](#_Toc188648217)

[Project Summary 5](#_Toc188648218)

[Project Objective 6](#_Toc188648219)

[Project Deliverables 9](#_Toc188648220)

[Non-Deliverables 12](#_Toc188648221)

[Assumptions 12](#_Toc188648222)

[Constraints 13](#_Toc188648223)

[Project Plan 15](#_Toc188648224)

[Gantt Chart 15](#_Toc188648225)

[Resource Sheet 15](#_Toc188648226)

[Team Organization 16](#_Toc188648227)

[Responsibilities 16](#_Toc188648228)

[Budget 18](#_Toc188648229)

[Change Management 18](#_Toc188648230)

[Progress/Status Reporting 20](#_Toc188648231)

[Decision Making 21](#_Toc188648232)

[Standards & Conventions 22](#_Toc188648233)

[Storage Units 22](#_Toc188648234)

[Executable & Source Code File Naming 22](#_Toc188648235)

[Backend Naming 22](#_Toc188648236)

[Frontend Naming 23](#_Toc188648237)

[Source Code Internal Module Header Documentation Content 23](#_Toc188648238)

[Database & Table Naming 23](#_Toc188648239)

[Variable / Function / Class Naming 24](#_Toc188648240)

[Backend: 24](#_Toc188648241)

[Frontend: 24](#_Toc188648242)

[Document Naming 24](#_Toc188648243)

[Note 25](#_Toc188648244)

[Appendices 25](#_Toc188648245)

[Appendix A: Team Contract/Charter 25](#_Toc188648246)

[Appendix B: Change Notification/Request Form 29](#_Toc188648247)

[Appendix C: Status Report Form 30](#_Toc188648248)

# Project Summary

**Summary:**

|  |  |
| --- | --- |
| Client | League Management Inc. |
| Project Name | League Management System |
| Project Acronym | LMS |
| Project Manager | Izaak Ford-Dow |
| Project Start Date | 2024-01-22 |
| Project Completion Date | 2024-03-21 |

**Definitions:**

|  |  |
| --- | --- |
| Guest | Someone that uses the system without elevated privileges or an account. |
| Member | A registered individual with specific system permissions based on their role, such as referee, coach, manager, scorekeeper, or parent. Members have more access and functionality compared to guests but do not possess full administrative privileges. |
| Staff | An employee that uses the system without full system privileges but have create, read and update privileges for jobs they have been assigned. |
| Admin | An employee that has full access to the system. |
| Database | An organized collection of the system data. |
| Company | League Management Inc. |

**Resources:**

|  |
| --- |
| Microsoft Office Software |
| Draw.io |
| System Analysis and Database Documentation (ERD, Class Diagrams, Use Case, etc.). |
| Business forms and processes documentation from the member |
| Human Resources (Project Manager, Programmer Analyst, DB Admin, Developer, Member, etc.). |

# Project Objective

The League Management Software (LMS) will be a comprehensive platform designed to streamline league operations for sports organizations of all sizes while addressing the needs of Guests, Members, Staff, and Admins.

* **For Guests:** LMS will provide public access for viewing league schedules, standings, and general information. Guests do not require an account and have no permissions beyond viewing publicly available data.
* **For Members:** LMS will enable individuals associated with a league—such as coaches, managers, scorekeepers, and parents—to access features based on their specific role. Members will have permissions tailored to their roles, such as updating team rosters (coaches), managing team schedules (managers), or recording match results (scorekeepers).
* **For Staff:** LMS will support employees working for the organization, granting them the ability to perform tasks such as editing website content, adding or removing teams, and managing league-wide settings. Staff will have create, read, and update permissions specific to their assigned responsibilities.
* **For Admins:** LMS will provide full access to administrators, enabling them to oversee all aspects of the system, including managing user accounts, league configurations, and sensitive data. Admins will have unrestricted permissions to ensure smooth operations and strategic oversight.

LMS will securely store all league-related information in an SQL relational database, ensuring that data is accessible only to users with proper credentials. Multi-factor authentication will safeguard user accounts and prevent unauthorized access.

Designed to support multiple sports and customizable scoring and ranking systems, LMS will be a scalable solution for sports organizations of any size. By integrating these features, LMS will optimize league management and deliver a user-friendly, secure, and efficient platform for all stakeholders.

This system will include the following functionalities:

* Guests can view public schedules, standings, and general league announcements.
* Members will perform tasks relevant to their roles, such as updating rosters (coaches), managing schedules (managers), and recording match results (scorekeepers).
* Staff can manage league operations, including editing website content, adding/removing teams, and maintaining league settings.
* Admins will oversee the system, manage user accounts, configure league settings, and generate detailed reports.
* Match results, player statistics, and league standings will update in real-time and be accessible to authorized users.
* Notifications and reminders will ensure timely actions, such as schedule updates, report submissions, and team-related tasks.
* Comprehensive reports will provide insights into team performance, league trends, and other key metrics.

This project is expected to be completed by March 21st, 2025, allowing time for user training and full deployment by July 15th, 2025. The budget for this project is $30,000.

# Project Deliverables

|  |  |  |
| --- | --- | --- |
| Phase | Deliverable | Description |
| **1** | **Elaboration & Preparation** | **Week 1** |
|  | Project Statement | Define the project’s scope, objectives, assumptions, constraints, and deliverables specific to LMS, such as league management, scheduling, and user roles. |
|  | Work Schedule/Gantt Chart | Develop a timeline for all project tasks and phases, ensuring milestones for user roles (Guest, Member, Staff, Admin) and their respective functionalities. |
| **2** | **Analysis** | **Weeks 2-3** |
|  | Stakeholder List | Identify and document all key stakeholders, including administrators, staff, coaches, players, and parents. |
|  | System Overview | Provide a detailed description of LMS, the deliverables (e.g., scheduling, statistics tracking, and notifications), assumptions, constraints, key requirements, and system modeling. |
|  | Subsystem Identification | Identify subsystems/modules such as user roles, scheduling, team tracking, and performance reporting. Assign these to the iteration schedule and refine as necessary. |
|  | Iteration-Specific Analysis | Perform event-based analysis, create use case models (e.g., "Coach updates roster"), domain class models (e.g., "Team," "Match"), and system sequence diagrams. |
| **3** | **Design** | **Weeks 4-5** |
|  | System Environment Design | Define the system environment, architecture, and deployment strategies using deployment diagrams. Include multi-platform support (e.g., desktop and mobile). |
|  | System Controls and Security | Define security measures, including role-based access control (RBAC), secure data storage, and multi-factor authentication. |
|  | Component Design | Create the ERD and data dictionary to define the database structure, covering tables for users, teams, matches, and performance statistics. |
|  | Database Design | Create the ERD and data dictionary to define the database structure, covering tables for users, teams, matches, and performance statistics. |
|  | User Interface Design | Develop storyboards or mock-ups showcasing user workflows, such as managing schedules, viewing statistics, and updating rosters. |
|  | Implementation Requirements | Document testing strategies, training needs, and staging environment specifications for iterative development and testing. |
| **4** | **Implementation** | **Weeks 5-9** |
|  | Source Code Development | Build and integrate system modules iteratively, adhering to coding standards. Complete key features like user management, scheduling, and statistics tracking. |
|  | Test Documentation | Create detailed test plans and test cases for functional, security, and performance testing. Document test results, ensuring issues are addressed promptly. |
|  | Deployment Deliverables | Prepare deployment instructions, including server setup, database migration scripts, and any environment configuration needed for system launch. |

# Non-Deliverables

League Management Inc. will provide its own hardware and network infrastructure to be used with the LMS system.

# Assumptions

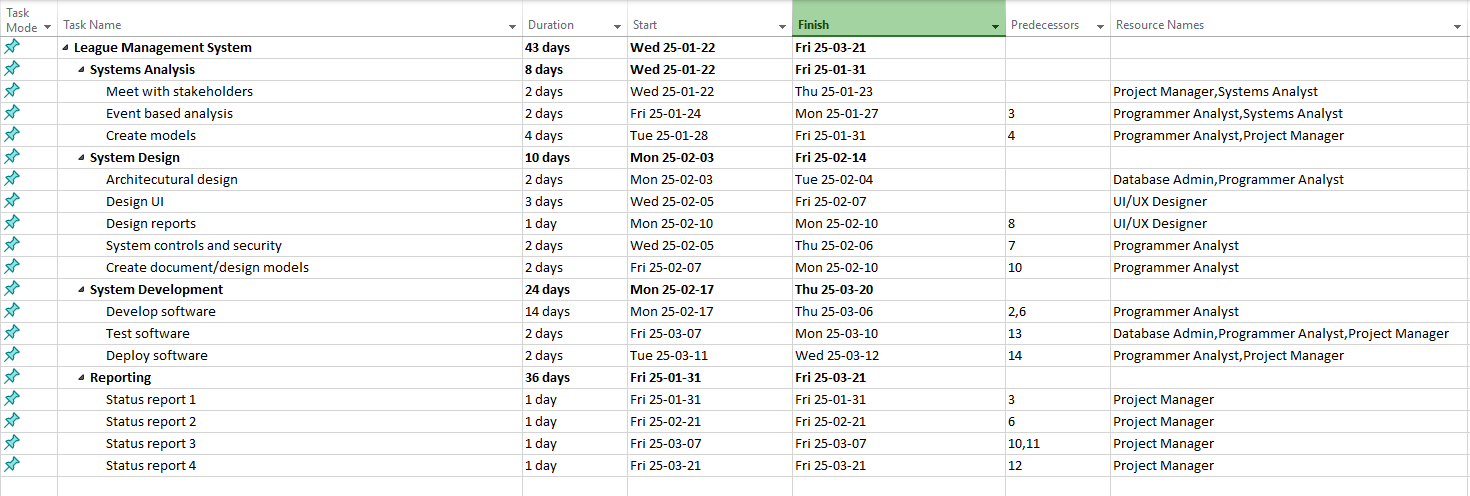
* Resource Availability: All necessary personnel, including developers, testers, designers, and stakeholders, will be available as needed throughout the project phases.
* Stable Requirements: The functional requirements for league management, including user roles, scheduling, and performance tracking, are clearly defined and unlikely to undergo significant changes during development.
* Existing Infrastructure: The necessary hardware and software for the development and deployment of the League Management System (LMS) are available and operational, including development environments and deployment servers.
* Stakeholder Involvement: Key stakeholders (administrators, staff, members, and coaches) will provide timely feedback during the requirements gathering, design, and testing phases.
* Data Accuracy: Player, team, and league-related data provided for the system is accurate and does not require extensive cleanup or transformation before use.
* Focused Scope: This project will focus on core functionalities such as user management, scheduling, and performance tracking, with additional features planned for future iterations.
* Budget Allocation: Sufficient budget and resources are allocated to cover the development, testing, and deployment phases.
* Technical Compatibility: The current technology stack, including databases, development frameworks, and tools, is compatible with the project requirements without requiring significant upgrades or replacements.
* Timely Testing and Feedback: System testing and user feedback will be provided promptly to meet the project timeline and ensure a smooth deployment.
* Security Standards: The project will adhere to existing security standards and regulations for protecting user data and ensuring system integrity.

# Constraints

* Fixed Timeline: The League Management Software (LMS) must be completed by March 21, 2025, to meet the planned deployment schedule for the upcoming league season.
* Budget Restrictions: The project must remain within the allocated budget, limiting the scope for extensive customization, third-party integrations, or additional resources.
* Compliance Standards: LMS must comply with relevant data protection and privacy regulations (e.g., GDPR) to ensure the secure handling of user and league data.
* System Integration: The software must integrate seamlessly with existing infrastructure, such as databases and other tools, without causing disruptions to ongoing league operations.
* Data Migration: Existing league, team, and player data must be accurately migrated or accessed within the constraints of current data structures.
* User Training: Limited time and resources for training require the system to be intuitive and user-friendly for administrators, staff, coaches, and members.
* Performance Requirements: The LMS must handle high volumes of real-time data, including match scheduling, ranking updates, and performance tracking, efficiently within its technical capabilities.
* Scalability Limitations: Initial deployment will focus on supporting a predefined number of leagues, teams, and players, with scalability improvements planned for future iterations.
* Resource Availability: Development and testing resources are limited, requiring careful planning and prioritization of features to ensure timely delivery.
* Security Measures: Strict security protocols must be implemented within the constraints of the existing technology stack to safeguard sensitive user and league information.

# Project Plan

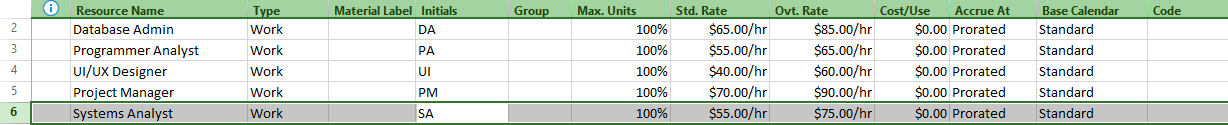
## Gantt Chart



A screenshot of a computer

Description automatically generated

## Resource Sheet



# Team Organization

A screenshot of a computer screen

Description automatically generated

## Responsibilities

**Project Manager**

* Oversees and plans the project's entire lifecycle.
* Coordinates with clients and stakeholders to gather input and feedback.
* Monitors and supports each stage of the project to ensure successful execution.
* Manages both time and budget to ensure the project stays on track.
* Reviews client feedback and evaluates change requests.
* Handles project documentation and ensures proper maintenance.

**Programmer Analyst**

* Examines existing systems and workflows.
* Identifies issues and areas for improvement.
* Defines the requirements for a new system to meet business needs.
* Develops a cost-effective system design to address identified problems.
* Documents the details and specifications of the proposed system.

**Database Administrator**

* Plans, installs, and upgrades database resources.
* Adjusts database structures based on input from developers and analysts.
* Optimizes database performance and monitors its functionality.
* Manages archived data to ensure it remains accessible and secure.
* Performs database backups and handles restoration as needed.

**UI/UX Designer**

* Collects and interprets system requirements to generate creative design concepts.
* Creates user interface elements and page layouts.
* Designs intuitive navigation systems.
* Identifies and resolves user experience issues.
* Refines designs based on user feedback and testing.
* Maintains consistent design standards across the system.

**Developer**

* Collaborates with team members to determine best practices and software requirements.
* Develops and writes efficient, reliable code.
* Modifies system software to fix issues, meet new requirements, or improve security and performance.
* Tests and evaluates newly developed components of the system.

**Client**

* Provides the project manager with detailed objectives, requirements, and authorizations.
* Approves any proposed changes to the system.
* Supplies the necessary funding for project completion.

# Budget

A screenshot of a graph

Description automatically generated

# Change Management

Change management involves the processes, tools, and techniques used to handle adjustments to the project to accommodate new or updated requirements. For the current project, change management and project management will work together to ensure the smooth transition from the current state through necessary changes and into subsequent iterations. While project management focuses on the technical tasks required to meet project goals, change management ensures all stakeholders are prepared for and supported during these transitions.

The project manager will be responsible for reviewing feedback from all stakeholders to assess the scope of any proposed changes. For minor changes that do not affect the overall scope, budget, or timeline of the project, the project manager will update the requirements and proceed without delay.

For changes that could impact the project's scope, budget, or deliverables, the project manager will meet with stakeholders to assess the new requirements and determine their implications. If the change is deemed necessary, a formal Change Request Form (refer to Appendix B) will be completed and signed by both the project manager and the stakeholders.

The Change Request Form will include the following elements:

* Key dates for decision-making and implementation of the proposed changes.
* A clear description of the proposed changes.
* An explanation of how the changes will affect the project’s timeline and budget.
* Recommended solutions or next steps.
* A section for approval or disapproval, including signatures from the necessary approvers.

This structured change management approach ensures that any revisions to the project are carefully considered, documented, and agreed upon by all relevant parties, minimizing disruptions and keeping the project on track.

Note: Refer to Appendix B for the Change Request Form template.

# Progress/Status Reporting

Project status reports will be prepared and provided according to the schedule below. Each report will be sent to all project stakeholders and the client to ensure transparency and keep everyone informed about the project's progress.

**Status Report Contents:**

* Project Overview: A summary of the current progress and milestones achieved during the reporting period.
* Issues and Recommendations: A detailed list of any problems or challenges encountered, along with proposed solutions or recommendations.
* Change Requests: Documentation of any requested changes to the project scope, timeline, or deliverables, with justifications and impacts.
* Next Steps: Plans and tasks for the upcoming work period to maintain progress and address outstanding issues.
* Budget Overview: A financial summary of the project’s budget status, including any variances, with the attached budget file for detailed review.

**Status Report Schedule:**

* Status Report One: February 2, 2025
* Status Report Two: February 16, 2025
* Status Report Three: March 2, 2025
* Status Report Four: March 16, 2025

Notes:

See "Appendix C – Status Report Form" for a template of the status report format to be used throughout the project. This ensures consistency and clarity in all reports.

# Decision Making

The decision-making process for the LMS project will follow a structured approach to ensure all key stakeholders, including the client, are involved and informed. The steps are as follows:

1. Identify the Problem: Clearly define the issue or challenge that needs to be addressed, ensuring all stakeholders understand its impact.
2. Gather Information: Collect relevant data, feedback, and insights related to the problem to inform the decision-making process.
3. Consult with the Client: Discuss the problem with the client and explore potential solutions, taking their input and priorities into account.
4. Evaluate Solutions: Assess the proposed solutions based on their feasibility, impact on the project timeline, budget, and overall goals.
5. Select the Solution: Aim to choose a solution that both the project team and the client can agree on, ensuring it aligns with the project's objectives.
6. Ongoing Discussion: If a solution is not selected, continue discussions until a resolution is found, prioritizing collaboration and mutual agreement.
7. Change Request Process: For any major changes to the project scope, a formal Change Request Form (Appendix B) will be proposed and reviewed by all parties.
8. Implement and Monitor: Once a solution is selected, implement it and monitor its effectiveness to ensure it resolves the issue as expected.

Note: If no decision is made within 24 hours, the client will have the final authority to determine the course of action to address the issue.

# Standards & Conventions

## Storage Units

The application will be developed on local computers (desktops/laptops) using XAMPP for the local development environment. XAMPP provides a consistent server environment with Apache, MySQL, and PHP for any required PHP-related tasks. The backend will be developed using Spring Boot for handling API-related operations and services. The frontend will be developed using React, and testing will be completed on a test server like the production infrastructure. The project will be deployed to production servers on the company’s infrastructure.

**Version Control:** Git will be utilized for version control of the source code, with the repository hosted on GitHub for source code and issue tracking.

## Executable & Source Code File Naming

### Backend Naming

**Executable Naming:** Backend executables for the Spring Boot application will be packaged as .jar files, following the format lms-[version].jar (e.g., lms-1.0.jar).

**Source Code Naming:** Source code files for the backend will be named using camelCase for Java files (e.g., UserService.java, MatchController.java).

### Frontend Naming

**Source Code Naming:** React components will use PascalCase for component names (e.g., PlayerList.js, TeamSchedule.js).

**CSS/SCSS Naming:** For styling, CSS and SCSS files will use kebab-case (e.g., player-list.css, team-schedule.scss).

**Assets Naming:** Images and other assets will follow kebab-case (e.g., background-image.png, logo-icon.svg).

## Source Code Internal Module Header Documentation Content

Each class or module will include the following header documentation:

* Author: Name of the module's author.
* Date: Date of module creation.
* Purpose: Brief description of the module's functionality.
* Change Log: Any updates or changes made to the module.

## Database & Table Naming

**Database Naming:** Databases will be named using PascalCase or initials of the project (e.g., LMS\_DB).

**Table Naming:** Table names will be in PascalCase and singular form (e.g., User, Team, Match).

**Column/Attribute Naming:** Column names will be singular and follow the pattern TableName\_AttributeName in PascalCase (e.g., User\_Email, Team\_Name, Match\_Score).

## Variable / Function / Class Naming

### Backend:

* Class Naming: Class names will follow PascalCase (e.g., UserService, MatchController).
* Variable Naming: Variable names will follow camelCase (e.g., userScore, playerId).
* Function Naming: Methods will follow camelCase (e.g., calculateMatchResults(), updatePlayerStatus()).
* Constant Naming: Constants will use UPPERCASE with underscores separating words (e.g., MAX\_PLAYERS, DEFAULT\_SCORE).

### Frontend:

* Class/Component Naming: React component classes will follow PascalCase (e.g., UserList, PlayerCard).
* Variable Naming: Variables in React components will follow camelCase (e.g., userData, matchInfo).
* Function Naming: Methods in React components will follow camelCase (e.g., handleClick(), updateScore()).
* Constant Naming: Constants in React components will follow UPPERCASE with underscores (e.g., MAX\_PLAYERS, DEFAULT\_LAYOUT).

## Document Naming

Documents will be named according to the following pattern, incorporating the project name, deliverable, and author’s last name and first name:

Format: ProjectName\_Deliverable\_AuthorLastName\_AuthorFirstName

Example: LMS\_ProjectStatement\_Smith\_John

Example: LMS\_DesignDoc\_Doe\_Jane

Note  
In addition to the previous standards and conventions, the project will adhere to the relevant coding standards and best practices outlined by the Spring Framework and React community guidelines. This includes maintaining consistency, readability, commenting, and maintainability across the codebase.

# Appendices

## Appendix A: Team Contract/Charter

**Member Contact Information**:

|  |  |  |
| --- | --- | --- |
| NAME | EMAIL | PHONE |
| Izaak Ford-Dow | [iforddow@hollandcollege.com](mailto:iforddow@hollandcollege.com) | 902-393-6781 |

**Communication:**

* Outlook
  + Text messaging
* Microsoft Teams
  + Video calls
  + Audio calls
  + Screen sharing
  + File sharing

**Goals:**

* Gain valuable experience that could potentially be used in future endeavours.
  + Meet project deadlines
  + Stay within project budget
  + Establish milestones
  + Learn from mistakes
* Successfully create and deliver the analysis, design and creation of the requested system ensuring that it is:
  + Functional
  + Usable
  + Efficient
  + Simple
* Marking goals?
  + Achieving a grade of 80% or higher would be considered optimal.

**Roles & Responsibilities:**

1. Izaak Ford-Dow – Project Manager/Project Lead (responsible for all roles)

**Meetings:**

* If the client or I need to discuss the project for an unplanned meeting, we can arrange to do so by using Outlook to plan it, then by using either the Microsoft Teams audio or video calling tool.

**Decision-Making:**

Decisions on the development of this project will be made by myself and the client. Ideally, both parties should agree that a decision is acceptable before moving on with the project. Decisions should be made in the following order:

1. Identify the problem/issue.
2. Gain all possible information on the problem/issue if necessary.
3. Discuss and evaluate solutions.
4. Attempt to choose a solution based on discussion.
5. If no decision is made, attempt discussing further.

Note: If no decision is made within 24-hours the client will have the ultimate say in which way to approach the issue.

**Team Project Contribution:**

This will be a solo project done by myself (Izaak Ford-Dow). I will be responsible for all work done and will reach out to my instructor if I have any questions or concerns.

**Conflict:**

If either myself or the client are to get involved in a conflict the following actions are to be taken:

1. Acknowledge the conflict.
2. Discuss the conflict and any possible resolutions for said conflict.
3. Attempt to resolve conflict in a manner that both parties are satisfied.

Note: If conflict persists after discussion, consider getting input from a third-party.

**Revisiting Charter:**

This charter will be re-evaluated using the decision-making protocols described in this Appendix. A review will occur at the midpoint of the project schedule.

Both the client and I will receive a signed copy of the original and revised charter.

## Appendix B: Change Notification/Request Form

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LMS | Change/Request Notification | | | | | | | |
|  | | | | | | | |
| DATE: | | | | REPORT #: | | | |
| TO: | | | |  | | | |
| FROM: | | | |  | | | |
| SUBJECT: | | | |  | | | |
| Key Dates | | | | | | | |
| Anticipated Decision |  | | | Expected Implementation: | | |  |
|  | | | |  | | | |
| Brief Description | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| Impact (Time) | | | | | Days: | | |
|  | | | | | | | |
|  | | | | | | | |
| Impact (Budget) | | | | | Amount: | | |
|  | | | | | | | |
|  | | | | | | | |
| Recommendation/Resolution | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| Approved | | Disapproved Reason: | | | | | |
|  | | | |  | | | |
| Approver Title | | | Approver Name | Signature | | Date | |
| Project Manager | | |  |  | |  | |
| Client | | |  |  | |  | |

## Appendix C: Status Report Form

|  |  |  |  |
| --- | --- | --- | --- |
| LMS | Progress Status Report | | | |
|  | | | |
| DATE: | | REPORT #: | |
| TO: | |  | |
| FROM: | |  | |
| SUBJECT: | |  | |
| Overview of Progress | | | |
|  | | | |
|  | |  | |
| Outstanding Issues/Recommended Action | | | Count: 0 |
| 1 |  | | |
|  | | | |
| Change Request Log | | | Count: 0 |
| 1 |  | | |
|  | |  | |
| Plans for Next Period | | | Count: 0 |
|  | | | |
|  | |  | |
| Overview of Budget (attached MS Project file) | | | |
|  | | | |