South Dakota School of Mines & Technology

Database Management Systems, Spring 2025

CSC 484-M01

Course Project – Requirements Document

Due
Wednesday, March 19th, 2025 at **11:59 PM** (Mountain Time).

Reference(s)
None

Software Required
None

Deliverable:

Submit a requirement document on D2L before the deadline. If you are working in a group, each member should submit the same document.

Use the template on the next page to create a database requirement document for a real-world application. The document should include essential details for both design and implementation.

- Background and application domain
- User and use cases
- Relationships
- Entities and related attributes
- Expected queries (not the final SQL statements, but frequently used search criteria and a rationale for them)

To ensure the database project get the appropriate size and complexity, your database requirements are supposed to meet the following objectives:

- Approximately 5 - 8 tables
- Approximately 5 - 8 use cases through UI (e.g. adding new record to table A, searching by name in table B, etc.)

Notes:

- Tasks for this project is done in a group of up to two students.
- Make sure you describe the project clearly.

South Dakota School of Mines & Technology

Database Management Systems, Spring 2025

CSC 484-M01

Course Project

XYZ System Requirement Document

Soccer Team Management

[Optional Logo]

(List names of participants)
Jacob Miklas, Mateo Clagg

1. INTRODUCTION

This document explains the requirements for the system to be created for CSC 484 Final Project. These requirements were created by student 1 (and student 2).

All questions, and complaints, can be directed to this (these) individual(s).
student 1: student1@email.com
student 2: student2@email.com
Jacob Miklas: jacob.miklas@mines.sdsmt.edu
Mateo Clagg: mateo.clagg@mines.sdsmt.edu

2. BACKGROUND AND APPLICATION DOMAIN

[In this section, clearly define the goals and functionalities of your database application. Provide enough detail for readers to grasp the core concept and features, ensuring they could replicate a similar application if needed.]
The goal of the program will be to help soccer clubs, coaches, and others manage teams, view player stats, etc. The system will store information such as team stats, player stats, matches, and possibly include information about referees.

3. USER AND USE CASES

[Outline the user(s) who will interact with your database application, detailing their interaction scenarios. For instance, a sales representative might utilize the system to process orders, while a cashier might employ it to handle payments.]

- coaches: manage teams, add/remove players, view team or individual players' statistics
- players: view team or individual statistics and look at team schedule
- referees: record new games and potential yellow and red cards
- league supervisors: schedule matches, update league standings, ability to manage teams

4. ENTITIES AND THE RELATED ATTRIBUTES

[Describe or tabulate the various entities that will be represented within your database, specifying the attributes that are essential for each. At this stage of conceptual design, feel free to use a range of data types like strings, integers, real numbers, and even arrays to effectively characterize these entities.

For example:

Entity	Attribute	Description/Remark
Car	Make (String) Model (String) Year (Integer)	A car is a thingy we ride in.

	Price (Float)	
Client	Firstname (String) Lastname (String) Email (String) Address (String)	A person that wants to buy a car.

]

Entity	**Attribute**	**Description/Remark**
Teams	TeamID (integer, PK) TeamName (string) CoachName(string) LeagueID (int, FK) Wins (integer) Losses (integer) Draws (integer)	A single team in a league.
Players	PlayerID (integer) FirstName (string) LastName (string) Age (integer) Position (string) TeamID (integer, FK)	Individual player on a team.
Matches	MatchID (integer, PK) Date (date-time) Location (string) HomeTeamID(integer, FK) AwayTeamID (integer, FK) RefereeID (integer, FK)	A match scheduled for the future.
MatchStats	MatchStatsID (integer) MatchID(integer, FK) HomeTeamGoals(integer) AwayTeamGoals(integer) PossessionHome(decimal) PossessionAway(decimal) YellowCardsHome(integer) YellowCardsAway(integer) RedCardsHome(integer) RedCardsAway(integer)	Stats after a match has been played.
PlayerMatchStats	PlayerMatchStatsID (int, PK) MatchID (integer, FK) PlayerID (integer, FK) Shots (integer)	Individual stats of a certain player.

	ShotsOnTarget (integer) GoalsScored (integer) MinutesPlayed (integer) YellowCards (integer) RedCards (integer)	
League	LeagueID (integer, PK) Name (string), City (string), Country (string)	Name of a league with a collection of teams.
Referee	RefereeID (integer, PK) FirstName (string) LastName (string) Experience (string)	Individual referees are able to referee assigned matches.

5. RELATIONSHIPS

[Describe the relationships existing among the various entities in your database. Also, specify any integrity constraints or rules that apply. For instance, you might describe a relationship such as "*Customers **acquire** cars*" and then define a rule like "*At any given time, a vehicle can be registered to only one owner*."]

Teams and Players:

- Foreign Key: TeamID
- Integrity Constraints:
    - A player must be assigned to a team
    - A team must have at least one player

League and Teams:

- Foreign Key: LeagueID
- Integrity Constraints:
    - Each team must be associated with exactly one league
    - A league must contain at least one team

Matches and Teams:

- Foreign Keys:
    - HomeTeamID -> TeamID
    - AwayTeamID -> TeamID
- Integrity Constraints
    - A match must involve exactly two distinct teams
    - A team cannot play against itself
    - A match must be schedules with valid team IDs

Matches and Referees:

- Foreign Key: RefereeID
- Integrity Constraints:
    - A referee cannot officiate two matches at the same time

Matches and Players:

- Foreign Keys:
    - MatchID
    - PlayerID
- Integrity Constraints
    - A player must be assigned to a match to have stats recorded

Matches and League:

- Foreign Key:
    - LeagueID
- Integrity Constraints:
    - A match must be associated with a league
    - A league must contain more than one match throughout a season

PlayerMatchStats and Players

Additional Integrity Constraints:

1. Primary keys:
    a. Each entity has a unique primary key
2. Foreign Keys and Referential Integrity
    a. All foreign keys must reference existing primary keys
    b. If a team is deleted, all its players should be deleted
3. Uniqueness Constraints
    a. No duplicate teams, referees, or players should exist
    b. A player's jersey number must be unique within team

6. EXPECTED QUERIES

[Outline the types of data retrieval operations (queries) you anticipate executing on your database. DO NOT WRITE SQL STATEMENTS.
For instance, "Generate a list of all clients who have overdue payments at intervals of 30, 60, or 90 days".]

- Retrieve team rosters and individual players
- Show top goal scorers and goal contributors
- View league standings
- See team schedules with past results
- Display games an individual referee has refereed