

## **1. Introduction**

### **1.1 SRS Purpose**

The purpose of this software is to serve as a college chess league tournament management system.

### **1.2 Product Scope**

The purpose of the college chess tournament management system is to ease tournament management and to create a convenient and easy-to-use application for tournament managers/administrators, trying to coordinate chess tournaments. The system is based on a local database with essential tournament management functions. We will have a database supporting up to 10 teams competing against one another weekly. Up to 3 sets of matches may be held on any one day a week. We hope to provide a comfortable user experience along with the best functionality available.

### **1.3 Intended Audience**

This project is a prototype for the college chess league tournament management system, and it is restricted within the college premises. The application is intended for use by tournament officials/administrators who will use the application to manage teams and develop a tournament schedule. This application is ***not*** intended for use by tournament participants as critical components such as scores and ranking are determined by the user of this application.

### **1.4 Definitions, Acronyms, and Abbreviations**

college: an institution offering instruction usually in a professional, vocational, or technical field

chess: a game for 2 players each of whom moves 16 pieces according to fixed rules across a checkerboard and tries to checkmate the opponent's king

tournament: a series of games or contests that make up a single unit of competition

management: the act or art of managing: the conducting or supervising of something (such as a business)

system: a regularly interacting or interdependent group of items forming a unified whole

application: a program (such as a word processor or a spreadsheet) that performs a particular task or set of tasks

### **1.5 Document conventions**

There are no document conventions.

### **1.6 References and acknowledgements**

Tsui, F. F., Karam, O., & Bernal, B. (2022). Essentials of software engineering. Jones and Bartlett Publishers.

Merriam-Webster.com. Merriam-Webster, 2011.

Web. 8 May 2011.

## **2. Overall Description**

### **2.1 Product perspective**

The database system stores the following information.

Team details:

It Includes team names and points.

Account details:

It includes usernames and passwords for access to management functions.

Schedule details:

It includes weekly information, tournament naming convention for the week, and matching information.

## **2.2 Product functionality**

The major functions are listed below along with a diagram.

Manage Teams. Add, Delete Modify.

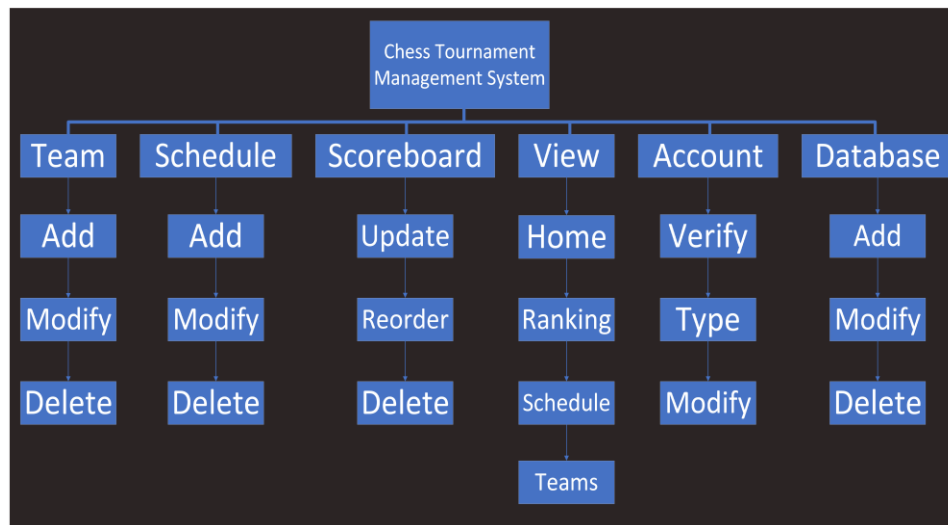
Manage Schedule. Add, Delete, Modify.

Manage Scoreboard. Update, Reorder, Delete.

Present information accurately and visually. Home page. Ranking page.  
Schedule page. Teams page.

Account verification and privileges. Verify, Multiple Types, Modify.

Store Information in a secure and accessible manner.



## **2.3 Users and characteristics**

The expected user shall be anyone with access to a computer which can run the JRE. There are no privilege levels/educational level/expertise nor experience required. The user is not expected to know Chess, as the rules and tips/strategies shall be included in the application. Users of the system should be able to retrieve tournament information from the database. The system will support two types of user privileges, Tournament Officials and Guests. Tournament Officials will have access to management and view functions, and the Guests will have access to view functions. The users will be expected to have basic knowledge of how to operate a computer. The

user will be expected to be able to use a keyboard and mouse. The user will be expected to be able to see clearly. The user will be expected to

## **2.4 Operating environment**

Operating environment for the college chess tournament management system is as listed below.

Operating system: Windows, Linux, Mac

Platform: Java 8+/Python 3

## **2.5 Design and implementation constraints**

Tournaments are to be: held weekly, on Saturdays, up to 3 matches (morning, evening, afternoon), 6 weeks long, may be extended by two weeks, each team should compete twice against each other team, shows which teams are playing each other each week, winner – 5 pts, loser – 1 pt, tie – 3 pts, shows scores highest to lowest, administrator and tournament officials have information entry access. Access to a computer is necessary.

## **2.7 Assumptions and dependencies**

It is assumed the user will have access to a computer and internet connectivity. The user will of the program will have access to Tournament Officials in order to access tournament information. Since this will be a standalone application, there shall not be any other dependencies aside from the requirement of having the Java environment installed in the computer.

## **3. Specific Requirements**

### **3.1 External interface requirements**

The application shall require no specific hardware other than a mouse for input, a monitor to display the application, and a computer that can support the JRE, which is needed to run the game files. A computer with basic

internet accessibility is required, this may include LAN port connection or Wi-Fi.

### **3.2 Functional requirements**

#### **Functional Requirements – What a program needs to do**

1. Display Teams – Show teams as text, illustrations, or both.
2. Store Teams – Keep data locally or remotely
3. Sort Teams – Winning, losing, tie, match date, ascending vs descending, numerical, alphabetical
4. Manage Teams – Up to 10 teams, add, delete, change
5. Take Input – Is input entered via CLI, GUI, or neither
6. Special cases, boundaries, or error condition – weekly, Saturdays, up to 3 matches (morning, evening, afternoon), 6 weeks long, may be extended by two weeks, each team should compete twice against each other team, shows which teams are playing each other each week, winner – 5 pts, loser – 1 pt, tie – 3 pts, shows scores highest to lowest, administrator and tournament officials have information entry access

### **3.3 Behavior requirements**

Seamless execution of actions when program is interacted with, and controls are pressed upon. Program behaves as the user would expect it to. Program shows responsive behavior and confirmation of actions.

## **4. Other Nonfunctional Requirements**

### **4.1 Performance requirements**

The application shall not crash. Other than that, there are not any specific performance requirements. The time complexity and space complexity have not been specified. The overall goal shall be to get the program running without any bugs or user-related problems.

The basic objective of performance is to reduce redundancy, which means that information is to be stored only once. Storing information several times

leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed, it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

#### **4.2 Safety and security requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

#### **4.3 Software quality attributes**

The software shall demonstrate maintainability, correctness, reusability, reliability, portability, and efficiency.

The signatures below indicate an agreement between parties to allow this document to serve as the baseline to begin development of the described software within.

CLIENT SIGNATURE

DEVELOPER TEAM 1 SIGNATURE

Software Engineering Team 1