

# **Technical Inspection Checklist**

For a Track & Field Meet Server

Version 2.0

Submitted in partial fulfillment of the requirements of the degree of MSE

Tracy Marshall

CIS 895 – MSE Project

Kansas State University

## **Table of Contents**

1. Introduction
2. Items to be Inspected
3. Technical Inspectors
4. Technical Inspection Checklist

## **1. Introduction**

The purpose of this document is to list the parts of the Track & Field Meet Server (TFMS) that need to be inspected. This document will guide the technical inspectors as they perform their inspection. They should use this document to assure coverage of the major components of the project. The technical inspection will be done to provide feedback on the correctness of the TFMS.

## **2. Items to be Inspected**

The items listed below will be inspected for consistency and correctness.

1. Executable Prototype

The inspection will act as a formalized buddy review also known as a peer review. More details about what will be reviewed are contained in the final section of this document.

## **3. Technical Inspectors**

- Blake Knedler
- Keith Moyer

## 4. Technical Inspection Checklist

Inspection Item	Pass/Fail/Partial	Comments
Code readability <ul style="list-style-type: none"> <li>- Did variable names make sense?</li> <li>- Did function names make sense?</li> <li>- Could you read through the code and understand the intent?</li> </ul>	PASS	Naming was very readable and made the use of comments not really needed to understand the intent of methods.
Code documentation and comments <ul style="list-style-type: none"> <li>- Were comments useful?</li> <li>- Did there need to be more comments?</li> </ul>	PASS	Code is readable, but not many comments explaining methods but not sure if they are really needed because code is understandable. Some code is commented out in TCPServer.py which should be removed. I would maybe add comments to getRandomID in TCPServer.py for magic numbers
Common coding practices <ul style="list-style-type: none"> <li>- Is duplicate code removed?</li> <li>- Are unused variables removed?</li> <li>- Are there any unnecessary imports?</li> </ul>	PASS	DataManager.py: class Dispatcher - unused response? - L: 26 Database.py: class Database - return True for user methods, unsure why returning anything. - L:54/61
Modular decoupled code <ul style="list-style-type: none"> <li>- Was the code structured in such a way to easily add and remove functionality?</li> </ul>	PASS	Yes, the code seems to be well architected and abstracted correctly with class structures Good use of inheritance in Users.py

## Track & Field Meet Server Technical Inspection Checklist

Code efficiency <ul style="list-style-type: none"><li>- Is there any unnecessary processing?</li><li>- Is there any processing that is done in the wrong place?</li></ul>	PASS	TCPServer.py: There is some code after your main, I would suggest putting that at the very top with other globals or in the main function.
--	------	---

**Inspected by: Blake Knedler**

**Signature: (Digital Sign) Blake Knedler**