[Company title and logo go here]

**Test Plan for**

**(Project name)**

Confidential and Proprietary Information of Datacard Worldwide

Contents

1.0 Introduction 3

1.1. Objective 3

1.2. Project Description 3

1.3. Process Tailoring 3

1.4. Referenced Documents 3

2.0 Assumptions/Dependencies 3

3.0 Test Requirements 3

4.0 Test Tools 4

5.0 Resource Requirements 4

6.0 Test Schedule 4

7.0 Risks/Mitigation 4

8.0 Metrics 4

Appendix A – Detailed Resource Requirements 5

Appendix B – Detailed Test Schedule 6

# Introduction

## Objective

## The Test Plan outlines the testing approach for GolfScore Release 1.1, focusing on verifying its functionality and performance.

## Project Description

## GolfScore aims to enhance the efficiency and accuracy of managing golf tournament data, providing tournament organizers with valuable insights and facilitating informed decision-making processes.

## Process Tailoring

The development process for GolfScore Release 1.1 involves tailoring standard software development and management processes to suit the specific requirements and constraints of the project. This tailoring ensures that the development activities align closely with the project's goals, timeline, and resource availability.

Types of testing

Specification Testing  
Functional Testing  
Limits Testing  
Stress Testing  
Performance Testing  
Document testing  
Network testing  
System testing

# Assumptions/Dependencies

* **Availability of Input Data:** It is assumed that accurate and complete input data files containing course records and golfer records will be provided for testing purposes.
* **Stable Development Environment:** The development team assumes a stable development environment with access to necessary tools and resources for coding, testing, and debugging.
* **Compliance with Golf Tournament Standards:** GolfScore assumes that the input data files provided adhere to standard formats and conventions for golf tournament data representation.
* **User Interaction:** The command-line interface assumes basic user proficiency in executing commands and interpreting textual output.
* **Error-Free Execution:** The software assumes error-free execution under normal operating conditions, with appropriate error handling mechanisms in place for exceptional scenarios.

**Dependencies:**

* **Code Completion:** The testing schedule depends on timely completion of code development and integration activities by the development team.
* **Prototype Availability:** Availability of a functional prototype is essential for conducting early testing and validation of key features.
* **Resource Allocation:** Dependencies exist on resource allocation, including personnel, testing tools, and equipment, to ensure timely completion of testing activities.
* **Input Data Availability:** Testing activities are dependent on the availability of input data files, which may be subject to production or simulation.
* **Documentation Completion:** The availability of comprehensive documentation, including software requirements specifications and design documents, is essential for guiding testing activities.

By acknowledging these assumptions and dependencies, the GolfScore project aims to mitigate risks and ensure smooth progression through the development and testing phases. Close attention to these factors allows for better planning and allocation of resources, ultimately contributing to the success of the project.

# Test Requirements

This section outlines the test requirements derived from the functional specifications and design of GolfScore Release 1.1. These requirements serve as the basis for creating test cases and procedures to validate the software's functionality and performance.

* **Input Data Validation:**

Verify that the program can handle various input file formats containing course records and golfer records.

Validate the parsing and interpretation of input data, ensuring accuracy and completeness.

* **Program Functionality Testing**

Test the execution of GolfScore via command-line interface (CLI) with different options and parameters.

Verify the generation of course reports with correct golfer details and stroke counts for each hole.

* **Error Handling and Recovery:**

Test the program's response to various error scenarios, such as invalid command-line options or missing input files.

Verify the program's ability to recover gracefully from errors and resume normal operation when possible.

* **Performance Testing**

Test the software's resource utilization under normal and peak load conditions to identify any performance bottlenecks.

* **Compatibility Testing:**

Verify the compatibility of GolfScore with different operating systems and environments, focusing on Windows 2000 and later version.

* **Documentation Validation:**

Ensure that the user documentation, including help information and report descriptions, accurately reflects the program's functionality and usage

# Test Tools

1. **Text Editors or IDE**

Text editors or integrated development environments (IDEs) will be used for creating and editing test cases, test scripts, and documentation related to testing activities. Examples include:

* Visual studio code
* Notepad++
* Eclips

1. **Command-Line Interface (CLI):**

Since GolfScore is a command-line application, a reliable CLI environment is essential for executing the software with different options and parameters. The native CLI of the operating system (e.g., Command Prompt in Windows) will be utilized for this purpose.

1. **Version Control System (VCS)**

A version control system will be employed to manage the source code, test scripts, and other project artifacts. This ensures version tracking, collaboration, and traceability of changes throughout the development and testing lifecycle. Commonly used VCS include.

* Git
* Subversion (SVN)
* Mercurial

1. **Test Case Management Tools**

Test case management tools help organize, execute, and track test cases, ensuring comprehensive test coverage and efficient test execution.

1. **Performance Testing Tools:**

Performance testing tools are essential for evaluating GolfScore's compliance with performance requirements and identifying potential performance bottlenecks. These tools simulate user load and measure system response times under various scenarios. Examples include

* Apache meter
* Load runner
* Gatling

1. **Compatibility Testing Tools**

Compatibility testing tools help assess GolfScore's compatibility with different operating systems, environments, and file systems. These tools automate the execution of tests across multiple configurations to identify compatibility issues. Examples include.

* Sause lab
* Browser stack
* Cross browser testing

1. **Documentation Tools**

Documentation tools assist in creating, formatting, and managing project documentation, including test plans, test reports, and user manuals. These tools enhance collaboration and ensure the accuracy and completeness of documentation. Examples include.

* Microsoft word
* Google doc

# Resource Requirements

The successful execution of the testing activities for GolfScore Release 1.1 necessitates adequate allocation of resources, including human resources, equipment, and other essentials. Below is an outline of the resource requirements for the testing process:

1. **Human Resources**

* **Test Engineers**
* **Test Manager**
* **Subject Matter Experts (SMEs)**
* **Documentation Specialists**

1. **Equipment and Tools:**

* **Computers**
* **Operating Systems:**
* **Version Control System (VCS)**
* **Test Management Tools**
* **Performance Testing Infrastructure**

1. **Training and Skill Development:**

**Continuous Learning:** Regular training sessions and workshops will be conducted to enhance the skills and competencies of test engineers, ensuring proficiency in testing methodologies, tools, and domain knowledge relevant to GolfScore testing.

1. **Communication and Collaboration:**

**Communication Channels:** Provision of communication channels (e.g., email, instant messaging platforms, project management tools) will facilitate seamless collaboration, information sharing, and coordination among team members, stakeholders, and external parties involved in the testing process.

# Test Schedule

The test schedule outlines the timeline and sequence of activities required to execute the testing phase effectively for GolfScore Release 1.1. Coordination with project stakeholders, development teams, and other relevant parties is essential to ensure alignment with the project's overall schedule and milestones.

# Risks/Mitigation

* Identifying potential risks and implementing mitigation strategies are essential components of effective project management. Below are some risks associated with the testing phase of GolfScore Release 1.1, along with corresponding mitigation measures:

1. **Resource Constraints**
   * + - **Risk:** Limited availability of skilled resources may lead to delays in test execution and insufficient test coverage
       - **Mitigation:** Allocate resources efficiently by prioritizing critical test activities.

Cross-train team members to handle multiple testing tasks and mitigate dependencies on individual experts.

1. **Incomplete Requirements:**

**Risk:** Ambiguous or incomplete software requirements may result in inaccurate test case design and ineffective testing

**Mitigation:** Collaborate closely with the development team and business stakeholders to clarify requirements.

1. **Unforeseen Technical Challenges:**

**Risk:** Unexpected technical issues, such as compatibility issues with third-party components or infrastructure limitations, may disrupt the testing process.

**Mitigation:** Perform thorough environment validation to identify and address potential technical constraints before test execution.

1. **Scope Creep**

**Risk:**

Changes to project scope or requirements during the testing phase may lead to scope creep, resulting in increased testing effort and timeline extensions

**Mitigation:**Implement robust change management processes to evaluate and prioritize scope changes based on their impact on testing

1. **Data Security and Privacy Concerns**

**Risk:**

Inadequate data protection measures may compromise the security and privacy of sensitive information used during testing

**Mitigation:**

Implement encryption and access controls to safeguard confidential data used in test environments.

# Metrics

Metrics are essential for evaluating the effectiveness of the testing process and assessing the quality of the GolfScore Release 1.1 software. The following metrics will be collected before and after product shipment:

**Prior to Shipment:**

1. **Effort Expended during Testing Phases**

Measure the total effort expended during Development Verification Testing (DVT), System Verification Testing (SVT), and Regression Testing. This metric provides insights into the resource allocation and testing efficiency.

1. **Number of Defects Uncovered**

Track the number of defects identified during DVT, SVT, Regression Testing, and the development phase. Categorize defects based on their severity and the testing phase in which they were discovered. Analyzing this metric helps in assessing the effectiveness of different testing activities and identifying areas for improvement.

1. **Test Tracking S-Curve:**

Plot the cumulative number of test cases executed over time to visualize the testing progress. The S-Curve provides a graphical representation of testing efficiency and helps in monitoring the test completion rate relative to the project schedule.

1. **PTR (Problem Trouble Report) S-Curve**

Similar to the Test Tracking S-Curve, plot the cumulative number of PTRs raised over time. This metric helps in tracking the resolution of reported issues and evaluating the effectiveness of defect management processes.

**After Shipment:**

1. **Number of Defects Uncovered Post-Shipment**

Monitor the number of defects reported by end-users or identified post-shipment. Analyze the root causes of these defects and assess their impact on the software's reliability and user satisfaction.

1. **Size of Software**

Measure the size of the software product in terms of lines of code, function points, or other relevant metrics. Comparing the size of the delivered product to initial estimates provides insights into project performance and helps in estimating future development efforts.

By collecting and analyzing these metrics, the project team can gain valuable insights into the quality of the GolfScore software, identify areas for improvement in the testing process, and make data-driven decisions to enhance product quality and customer satisfaction. Regular monitoring and reporting of these metrics throughout the project lifecycle ensure proactive management of quality-related issues and facilitate continuous improvement initiatives.

Appendix A – Detailed Resource Requirements

[To estimate the resource, all test activities must be identified and resources needed to accomplish the activities estimated. Detailed estimates will be shown here. This consists of identifying all project test activities by the Test Group and the number of hours estimated to accomplish these activities. Be specific. Show specific responsible test engineer’s names, if possible. A grand total of the effort must be shown here, as well as in Section 5.0.]

Appendix B – Detailed Test Schedule

[Attach two charts, viz. Gantt and PERT. In Gantt, main activities are shown as a list on the Y-column with bars parallel to the X-axis, showing the timeframe to perform activities. In PERT, dependencies of each activity must be identified.]