**CS 415 Software Testing**

12/17/2012

**Test Plan**

**Game Connect**

**Joaquim Costa**

Table of Contents

[1. Revision History 2](#_Toc343370951)

[INTRODUCTION 3](#_Toc343370952)

[1.1 Objectives 3](#_Toc343370953)

[1.2 Scope 3](#_Toc343370954)

[1.3 Reference Material 3](#_Toc343370955)

[1.4 Definitions and Acronyms 3](#_Toc343370956)

[2. TEST ITEMS 5](#_Toc343370957)

[3. FEATURES TO BE TESTED 5](#_Toc343370958)

[4. FEATURES NOT TO BE TESTED 5](#_Toc343370959)

[4. APPROACH 6](#_Toc343370960)

[4.1 Component Testing 6](#_Toc343370961)

[4.2 Integration Testing 6](#_Toc343370962)

[4.3 Security Testing 6](#_Toc343370963)

[4.4 Regression Testing 6](#_Toc343370964)

[4.5 Acceptance Testing 7](#_Toc343370965)

[5. PASS / FAIL CRITERIA 7](#_Toc343370966)

[5.1 Suspension Criteria 7](#_Toc343370967)

[5.2 Resumption Criteria 7](#_Toc343370968)

[5.3 Approval Criteria 7](#_Toc343370969)

[6. TESTING PROCESS 7](#_Toc343370970)

[6.1 Test Deliverables 7](#_Toc343370971)

[7. ENVIRONMENTAL REQUIREMENTS 7](#_Toc343370972)

[7.1 Hardware 7](#_Toc343370973)

[7.2 Software 8](#_Toc343370974)

[7.4 Tools 8](#_Toc343370975)

[7.6 Risks and Assumptions 8](#_Toc343370976)

[7.6.1 Assumptions 8](#_Toc343370977)

[7.6.2 Risks 8](#_Toc343370978)

[8. PLAN APPROVALS 9](#_Toc343370979)

# 1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Revision Date | Description of Change | Author |
| 1.0 | 12/15/12 | First draft of the test plan | Joaquim Costa |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# INTRODUCTION

The Test plan has been created to communicate the test approach to the team members; it includes the objective, scope, risks, approach… (Refer to the table of content).

## 1.1 Objectives

Game Connect is a client/server application that will allow user to play multiplayer game online. The server will be responsible for handling logins, keeping track of players rating and account information through a MySQL database, and connecting user for online multiplayer game.

Phase 1 of the project will deliver the Game Connect with functionality of:

* Creating user account
* Allow user to view stats. Info.
* Allow user to initiate an online multiplayer
* Allow user to chat while playing
* Ability to choose to play either Checkers or Chinese Checkers

## 1.2 Scope

The initial phase will include all requirements that are necessary for a multiplayer game connection. At the end of this phase the program should be able to allow a user to:

* Create an account
* Initiate an online game with another player and chat
* Select to player either Checkers or Chinese Checkers
* View status of win/loss

## 1.3 Reference Material

* Requirements Specification for the Game Connect System
* Requirements Specification for the Checkers Game

## 1.4 Definitions and Acronyms

|  |  |
| --- | --- |
| **unit testing** | The smallest test level where we test individual objects/classes methods. This level of testing is mainly concerned with syntax errors and compiles problems |
| **integration testing** | This is the next level up from unit testing. Integration testing tests how two objects interact with each other and makes sure they are working correctly. |
| **System testing** | This is basically the final level of testing. This is when we put everything together and use it to make sure that it’s working correctly. We will look back at our requirements at this stage and make sure that our program meets them correctly. |
| **acceptance testing** | This is the final level and something that we may not even have to do or even have time to do. This level is concerned with our programs appeal to users. We will let regular people use the program and record what they like and dislike about it. |
| **white box** | testing where we know the implementation details of the program (we’re looking at the code) |
| **black box** | Testing when we don’t know the implementation details (we are using the program but not looking at the code). |
| **test basis** | What we are using to test against and what we are testing (how we know if something is working correctly or not.)  Meaning I can be testing a piece of code that implements one of the requirements. The code and the requirement are my basis because I am testing to make sure the code adheres to the requirement. |
| **exploratory testing** | This is where we try to break our system. We write a quick plan on what we’re going to try, then spend about 20 minutes or so trying to break the program afterwards just write a quick thing (paragraph maybe) or what we did. |
| **Packet sniffing** | packet sniffer captures all of the packets of data that pass through a given network interface |
| **Wireshark** | Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education |
| **HTTPS** | Hypertext Transfer Protocol Secure (HTTPS) is a widely used communications protocol for secure communication over a computer network. |
| **SSL** | Secure Sockets Layer (SSL), are cryptographic protocols that provide communication security over the Internet |

# 2. TEST ITEMS

* User Account
  + Create account, update user info, view stats history
* Multiplayer
  + Make connection with another player, initiate online game, end a game, chat
* Database Services
  + Make connection with the database, backup database, restore database
* Checkers Game,
  + The board, win/loss, moves, pieces, king
* Chinese Game

From the test items named above I consider that User Account, Multiplayer, and Database Service are the most critical. They should all have the same priority because they are a fundamental part of the Game Connect system.

The Checkers Game and the Chinese Game are also important, but obviously not as important as the other one that we have mention earlier, primarily because they can be a standalone application on its own. Nevertheless, we still have to test them to ensure we deliver a working game to the customer.

# 3. FEATURES TO BE TESTED

* User Account
* Multiplayer
* Database Services
* Checkers Game
* Chinese Game

# 4. FEATURES NOT TO BE TESTED

* Single Player – This is low risk suite we will implement it but not test it, by testing the multiplayer suite most of this suite will be explicitly tested as anyway.
* P2P files sharing – this features might not even be implemented
* Player vs. Artificial Intelligence (AI) – this feature might not be implemented as well.
* Other games example: Blackjack, Tic Tac Toe and Connect Four.

## 4. APPROACH

We will use a Process or standard-compliance approach in this project. The project is using an agile methodology, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

For the test levels of unit and integration we will be doing white box testing. We will be using the source code itself and our documentation as a test basis. For the test levels of acceptance we will be doing black box testing using our program, the requirements document and our use cases as a test basis. The system testing will be supplemented with individual exploratory testing and acceptance testing will be supplemented with user testing.

We will use the traceability matrix to ensure that all the features to be tested have been tested.

These test levels will be documented and be a part of the testing deliverables.

## 4.1 Component Testing

Component testing will be performed automated. The unit test software will be integrated with the Test Manager software. It will use the input and expected value acquired from the test manager to validate if a test passed or failed. The test will be completed by running the unit test software.

## 4.2 Integration Testing

Our Integration test will be done incremental rather than “big bang”. Integration testing will be performed after the all the component tests are done. Here we will make sure that all the games work well within the system, the database, user account, and the multiplayer all communicate with one-another. We will ensure that the design objectives are met; the software feature works as a complete entity and it complies with operational requirements.

## 4.3 Security Testing

We will do exploratory testing and try to exploit the system through the multiplayer feature. We will use networking tools such as Wireshark and packet sniffing to help us find vulnerabilities in the multiplayer subsystem.

## 4.4 Regression Testing

Whenever changes are made to the code we should check the traceability matrix to see if the changes might affect other component; if the change in the code affects other components we should run a unit test in the affected components to ensure that applied changes to the application have not adversely affected previously tested functionality.

## 4.5 Acceptance Testing

When our program is in its final phase we will invite users to test it before we launch the software.

# 5. PASS / FAIL CRITERIA

In order for an item to pass 95% of that items test case must pass, in addition the 5% that failed must all be of LOW impact risk. Otherwise the project cannot move on to the next phase until the developers fix the code.

## 5.1 Suspension Criteria

At the end of each day (4:00pm) testing will be suspended. At that time, all test cases executed during the day should be marked and saved. At the end of each day the test team should perform a backup of all the work they did for that day.

## 5.2 Resumption Criteria

Testing should continue from the point of last suspension. When all the testing has been performed a report will be printed.

## 5.3 Approval Criteria

A testing is approved when the output is equal to the expected output.

# 6. TESTING PROCESS

## 6.1 Test Deliverables

* Test Plan
* Test Design
* Traceability Matrix

# 7. ENVIRONMENTAL REQUIREMENTS

For the testing environment we will need at least two computers so that we can test the multiplayer feature of the software. The computers must be connected to a LAN (Network) and have connection to a printer so that testing reports can be printed. The computers must have HTTP Server and MySQL Server installed. Also, the Game Connect software must be available in both computers.

## 7.1 Hardware

* At least 2 computers
* Network Access (LAN)
* Printer
* Mouse and a Keyboard

## 7.2 Software

* Game Connect software
* Java Runtime Environment (JRE)
* A server that supports Java (i.e. Apache Tomcat).
* MySQL Server

## 7.4 Tools

* Wireshark
* Packet Sniffing
* Test Manager
* Automated Unit Tool (JUnit)

## 7.6 Risks and Assumptions

### 7.6.1 Assumptions

This section lists the assumptions that are made specific to the project.

* We assume that this application will be run on multiple platform (i.e. Windows, Linux, and Mac)
* We assume main programming language used in this project is Java.

### 7.6.2 Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk** | **Impact** | **Trigger** | **Mitigation Plan** |
| 1 | Intrusions to users privacy trough the Game Connect Application | High | Hackers exploit vulnerability in our application and get access to users computer | Implement the system using HTTPS or SSL |
| 2 | Cannot Integrate Games into the Game Connect | High | The game connect interface is different than other individual games, so it won’t integrate | Define an interface that should be implemented by all games that want to integrate with the Game Connect. Integration with other games will be as easy as plugins now. |
| 3 | The database gets corrupted | High | The server goes down abruptly which caused the database to get corrupted losing as information | Develop a subsystem that does database backups and restores automatic. |
| 4 | Lead Tester, Programmer,… abandons the project | High | One of the lead person quits the jobs | Have a lead number two for each role that would be able to handle this situation. |

# 8. PLAN APPROVALS