# PROJECT REVIEW III

# VERIFICATION, VALIDATION AND PERFORMANCE ANALYSIS

## of

## <<Project Title>>

## Approved

<<Internal Guide>>

## Prepared By

<<Name of the student>>

<<Regno>>

# TABLE OF CONTENTS

**Contents** **Page No**

**Appendix**

# 1. INTRODUCTION

## 1.1 Overview of the Project

**<<***A brief overview about the project should be specified including the scope of the project, and its features>>*

## 1.2 Document Conventions

*<<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority>>.*

## 1.3 Motivation & Background

*<<Mention about the key characteristics of the project, what made you to take up this project, what are the prerequisites of the project and what results are expected (the desired outcome) at the end of successful completion. A description of what is expected to be done within the project, and how to produce the expected amount of work should be mentioned. This is to be planned prior to the implementation process to make a foundation for further goal setting and implementation>>*

## 1.4 Problem Statement

*<<A problem statement is a statement of a current issue or problem that requires timely action to improve the situation. A summary of the negative points (problems) identified in the existing systems should be mentioned. The problem proposed to be addressed should be clearly specified. The problem statement should concisely explain the barriers faced by the current systems. The statement should be completely objective, focusing only on the facts of the problem and leaving out any subjective opinions. >>*

## 1.5 Objective of the Project

*<<The objective of the proposed project should give a comprehensive view of the project to the user. The proposed solution to the existing problems mentioned in 1.4, scope and goals should be made clear in this section.>>*

## 1.6 Project Scope

*<<Project scope specifies the boundary of the entire project. It should specify the required resources, deliverables, features and tasks within a given project. It should outline the entire project, including any deliverables and their features, as well as a list of stakeholders who will be affected. It will also include any major project objectives, deliverables and goals to help measure success. Start your project scope statement by explaining the need for your project, and how the end result will solve that need. List out what is within the scope of your project, and what is out of scope. This will help establish boundaries for the project to exist. Assumptions and constraints may also be mentioned. A context diagram may be accompanied to depict the scope of the system and how it would relate and interact with other systems.>>*

## 1.7 References

*<List any other documents or Web addresses to which this document refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location>*

# 2. VERIFICATION AND VALIDATION

## 2.1 Component Testing

*<< Component testing is a software testing type, in which the testing is performed on each individual component separately without integrating with other components. It's also referred to as Module Testing when it is viewed from an architecture perspective. Component Testing is also referred to as Unit Testing, Program Testing or Module Testing. Generally, any software as a whole is made of several components. Component Level Testing deals with testing these components individually. It's one of most frequent black box testing types which is performed by QA Team.>>*

## 2.2 Component Test Plan and Strategies

*<<Describe the components and the test plans and strategies used for testing the component>>*

### 2.2.1 Component 'n' <<Name of the Component>>

#### 2.2.1.1 Functions to be Tested

*<<Test Case Group : Describe where in the system structure and the overall test plan identification this particular test case group fits.>>*

#### 2.2.1.2 Testing Approach

*<<Describe the functions and combinations of functions that are being tested. These features should be traced to the functional or non-functional requirements for confirmation; this should be related to the information given in the test-requirements cross-reference matrix.>>*

*This should describe:*

- *The specific test strategies and techniques to be used on this test case group*
- *The methods that will be used to analyze the results*
- *The rationale behind the choice of the test cases*

#### 2.2.1.3 Pass/Fail Criteria

*<<Specify the detailed criteria that the product must satisfy in order to pass this test case group.>>*

#### 2.2.1.4 Individual Cases

*<<This should include the following information :>>*

- *Test case identifier:*
- *Input:*

    - *Specify in terms of ranges of values, input protocols, and structure.*
    - *Give details about all files, databases, etc. that are required to support the inputs.*
- *Expected output*
    - *Specify the correct output for the given inputs.*
- *As much as possible, the expected output should be given in sufficient detail to allow the outcome of the test case to be simply "pass" or "fail".*
- *Environment*
- *Describe any special test environment requirements, including software, hardware, specially trained personnel, and monitoring equipment.*
- *Special procedures*
- *Describe any unique procedural requirements Precedence and dependencies*
- *Describe the order in which test cases must be run (if this matters) and any interdependencies between cases.*

*<<Repeat for all test cases>>*

*<<Add all testing of all components>>*

## 2.3 System Integration Testing

### 2.3.1 System Test Strategy Overview

*<The entire software product is tested after each individual component has already been tested. The integration tests ensure that the components work together as well as they works individually>*

### 2.3.2 Linking Integration

*<Test Case Group: Linking Integration*

*Features Tested: Links and buttons from the various HTML pages within the game Testing*

*Approach: Inspection*

*Pass/Fail Criteria: All modules must be functioning as tested individually in the component verification. The majority of the links will be included in the functional test components of the previous section>*

## 2.4 Acceptance Testing

*<Before the product is presented to the client, it will be acceptance tested in the following manner: Limited usability testing will be performed by a group of stakeholders. A prototype will be shown and evaluated. Other usability testing will be performed regularly by all members of the team.*

*The components at the highest levels of the system will be tested first. Each component will be followed to a sub-component and then tested until the most detailed components at the lowest levels have all been tested and any errors have been corrected. In this way all possible user scenarios are considered.>*

### 2.4.1 Specific Acceptance Criteria

*<Mention the specific criteria under which the project will be accepted by the client. Keep your criteria well-defined so any member of the project team understands the idea you're trying to convey. Keep the criteria realistic and achievable. Define the minimum piece of functionality you're able to deliver and stick to it. On the other hand, don't try to describe every detail since you risk cluttering up your backlog and getting buried under many small tasks. Coordinate with all the stakeholders so your acceptance criteria are based on consensus. Create measurable criteria that allow you to adequately estimate development time so you're able to stay within budget and time constraints. Consider providing checklists that enable you to see what user stories are covered with acceptance criteria.>*

**Sample**

*The product should run from the S2S website.*

*The product should run on multiple platforms. All tests on the product should succeed on popular platforms such as Windows, and Macintosh.*

*The product will have a colorful graphical user interface.*

*The product should receive input from the user and the user will be directed to the correct page and information.*

*The product should display scores to evaluate the performance of the user after he/she has completed the game.*

*The product will be tested for invalid inputs>*

## 2.5 Installation Procedure

*<<Describe the deployment and installation procedure for the software>>*

*Sample*

*<The product can be accessed via the Internet website so no installation is required. However, Flash plug-in must be installed>*

# 3. RESULTS AND DISCUSSION

## 3.1 Analysis and Inference of the Results

*<< The purpose of a Results section is to present the key results of your research/project. Present your results as figures and tables and point the reader to relevant items while discussing the results. This section should highlight significant or interesting findings. Be sure to include negative results and highlight potential limitations of the project. Charts may be included for presenting the results.>>*

## 3.2 Performance Metrics

*<Specify the metrics used for analyzing the performance of your system.>*

## 3.3 Performance Analysis with Existing Systems

*<Compare your system with existing systems and analyze the performance of your system in terms of the metrics. The results may be presented in the form of charts or tables.>*

# 4. CONCLUSION AND FUTURE SCOPE

## 4.1 Conclusion

## 4.2 Future Scope

# Appendices

A. Test Screen Shots
B. Result Analysis Charts
C. Performance Analysis Charts