SE 3XA3: Test Report
Title of Project

Team #, Team Name
Student 1 name and macid
Student 2 name and macid
Student 3 name and macid

December 6, 2016

Contents

1 Functional Requirements Evaluation 1

2 Nonfunctional Requirements Evaluation 1

2.1 Usability 1

2.2 Performance 1

2.3 etc. 1

3 Comparison to Existing Implementation 1

4 Unit Testing 1

5 Changes Due to Testing 1

6 Automated Testing 1

6.1 Automated Testing Strategy 1

6.2 Specific System Tests 1

7 Trace to Requirements 2

8 Trace to Modules 2

9 Code Coverage Metrics 2

List of Tables

1 **Revision History** i

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
Date 1	1.0	Notes
Date 2	1.1	Notes

This document ...

1 Functional Requirements Evaluation

2 Nonfunctional Requirements Evaluation

2.1 Usability

2.2 Performance

2.3 etc.

3 Comparison to Existing Implementation

This section will not be appropriate for every project.

4 Unit Testing

5 Changes Due to Testing

6 Automated Testing

6.1 Automated Testing Strategy

For this project we elected not to use a 3rd party testing library. We made this decision to ease configuration/installation problems and reduce our dependencies, as we judged it would not be necessary. Instead a series of files (labeled test.foobar.cpp) in the repository hold tests, which are run by our custom test runner. These automated tests are run on command by executing the produced executable, or by the continuous integration script run whenever changes are pushed to the central repository. The results of these tests are automatically reported, resulting in a failed or successful build.

6.2 Specific System Tests

The following is a list of all system tests in the project.

Name:	Test 1
Initial State:	jaldsjfk
Input:	aksdlfjlad
Expected Output:	akdslfjadjsf

7 Trace to Requirements

8 Trace to Modules

9 Code Coverage Metrics