



Test Plan

Table of Contents

1 Introduction	2
1.1 Objectives	2
1.2 Team Members	2
2 Scope	2
3 Assumptions / Risks	3
3.1 Assumptions	3
3.2 Risks	3
4 Test Approach	3
4.1 Test Automation	4
5 Test Environment	4
6 Milestones / Deliverables	4
6.1 Test Schedule	4
6.2 Deliverables	4

1 Introduction

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

1.1 Objectives

The goal of this test plan is to verify functionality, check that requirements are met, and ensure the final product is robust and bug-free.

1.2 Team Members

Resource Name	Role
Jayde Holbrook	Scrum Master, Backend Dev
Ben Lukins	Test Lead, Backend Dev
Patrick Berzins	GUI Lead, Frontend Dev
Nathaniel Smith	Technical Lead, Backend Dev
Van Hudson	Technical Writer, Frontend Dev

2 Scope

The final delivery will include all ‘must have’ requirements. These and any other requirements that get included must all be tested. Upon completion of the software, a tester must be able to:

1. Select a performance to watch
2. Choose as many available seats as desired
3. Check out
4. Exchange purchased tickets for new ones
5. Purchase a season ticket
6. View all saved data
7. Add performances
8. Add purchases as an admin
9. Update existing purchases when they’re paid for or picked up
10. Edit default section prices
11. Export the list of season ticket holders
12. Import an existing list of season ticket holders

Load testing will not be considered part of this project since the service is being hosted locally.

Importing/exporting data other than the season ticket holder list is not considered within the scope of this project.

3 Assumptions / Risks

3.1 Assumptions

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

1. It is assumed that users would be emailed their confirmation number so that they are able to enter it in order to exchange tickets

No other specific assumptions were made during the testing of this project other than standard assumptions such as that the program launches successfully

3.2 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered.

- Team Member Drops Out/Gets Fired: 8
 - o Verify that everyone would like to graduate
- COVID: 9
 - o Wash your hands and stay home if sick
- Class/work conflicts: 3
 - o Communicate when we are and aren't available
- Member(s) not learning required tools: 8
 - o Using online resources to learn
- Technology breaks down: 8
 - o Using well-known and reliable resources and redundancy of personal technology
- Platforms do not work together: 7.5
 - o Proper research beforehand
- Platforms require payment to use: 8
 - o Proper research beforehand. Otherwise, research alternate platforms
- Software management indicators, including indicators to be used (i.e., metrics) through weekly individual and team reports
- Security and privacy
 - o Two factor authentication
 - o Don't share passwords
 - o Only ticket sellers may see, access, and export ticket buyers' personal information.

4 Test Approach

As GitHub branches and pull requests were used extensively throughout the development process, a large segment of testing was code reviews initiated by the pull request process. Whenever a segment of code was deemed finished by the person who wrote it, multiple other team members tested that new functionality and ensured that no existing functionality was compromised.

When the project was nearing completion, detailed and thorough test cases were devised to test all functionality and ensure that all requirements were met while also providing a robust and responsive application.

4.1 Test Automation

No automated unit or functional tests were used in the development of this application.

5 Test Environment

The application will be tested on computers running Windows 10/11, Linux, and macOS.

6 Milestones / Deliverables

6.1 Test Schedule

Task Name	Start	Finish	Comments
Test Planning	2023/01/30	2023/04/28	
Review Requirements documents	2023/01/15	2023/01/30	
Create initial test estimates	2023/01/15	2023/01/30	
Functional testing – Iteration 1	2023/04/01	2023/04/18	Up to demo
Functional testing – Iteration 2	2023/04/18	2023/04/28	Post-demo
System testing	2023/04/19	2023/04/28	
Regression testing	2023/04/23	2023/04/28	
UAT	2023/04/23	2023/04/28	
Resolution of final defects and final build testing	2023/04/27	2023/04/28	
Performance testing	2023/04/17	2023/04/19	Scaled up to 20k purchases
Release to Production	2023/04/28	2023/04/28	

6.2 Deliverables

Deliverable	For	Date / Milestone
Test Plan	Project Manager; QA Director; Test Team	2023/04/28
Test Results	Project Manager	2023/04/28
Metrics	All team members	2023/04/28