Software Test Plan

Project: Self-Teaching Tool for Concepts in Number Theory

Introduction

1. Purpose

The purpose of this document is to outline the testing plan for a self-teaching visualisation tool for concepts in number theory.

2. Project Overview

My dissertation project is to develop an interactive visualisation tool that allows individuals and groups to solidify their understanding of concepts in the field of number theory. These concepts build towards a visual explanation of the RSA cryptosystem, with related (but not necessarily central) ideas explained/visualised along the way.

The tool (currently in development) is a web-based application that users can easily access through any modern browser, and has a simple page structure that follows a logical progression through the material taught.

The Visualisation Tool for Concepts in Number Theory currently has one screen to demonstrate the Additive Structure of a Z_n group, and upon project completion will have multiple screens.

3. Audience

This product is intended for students or teachers in higher education for fields directly involving or adjacent to pure mathematics. The visualisation tool can be used by individuals to develop a deeper intuition for how concepts in number theory interact and fit together, but it can also be used by educators in a group setting to aid interactivity in classes.

Test Strategy

1. Test Methodology

The testing process outlined in this document describes the implementation of Agile methodologies via iterative testing and continuous integration. This test plan also adopts User Acceptance Testing and Test-Driven Development approaches, as the user experience dominates the requirements in the corresponding Software Requirements Specification (SRS) so integrating user feedback with the development process maximises efficiency.

2. Test Environment

The environments in which the product will be tested are to be deliberately varied, to assess how the system meets liveness, compatibility, and availability requirements. The environment configurations will be varied across devices, operating systems, and browsers.

3. Test Objectives

Unit Testing

Tests will be performed on individual components of the tool to validate and verify functioning. In the scope of this product, a component will be considered an individual screen.

Integration Testing

Integration tests will verify that the software behaves as expected independent of the testing environment and that the content management requirements specified in the SRS are met.

System/User Acceptance Testing

Tests performed by a representative group of end-users will be repeated regularly along the development process of the system to ensure that the tool meets user expectations and requirements.

4. Levels of Testing

Functional Testing for Liveness

Test Acceptance Criteria

The tool can be used on various devices, operating systems, browsers, and combinations thereof.

User Acceptance Testing

The following tests assess fulfillment of all remaining functional and non-functional requirements.