Project Thor

Test Results Document

Version 1.0

Team Members:
Adonay Pichardo
Jared Blanco
Josh Temel
Luke Bonenberger

Faculty Advisor:
Ph.D, Sid Bhattacharyya

Client:
Ph.D, Amitabh Nag

Florida Institute of Technology Senior Design

Mar 19, 2022

Table of Contents

1. Introduction

This Test Document is a document to provide documentation to be used in the aid of testing Project Thor. Within are test cases that check to see if the project conforms to it's requirements and client needs.

1.1. Purpose

In the previous documentation for Project Thor the project's design, goals, features, client needs and requirements have been specified. The purpose of this document is to verify that the project has been developed to meet the requirements set forth for it in the Requirements Document as well as validate that it meets the customer's needs as discussed in the Project Plan Document.

1.2. Scope

As stated in the purpose section, this document is provided to ensure the verification and validation of Project Thor. It does so through specifying specific test cases designed for just that purpose. However, not every possible test case can be run for every possible input for this project. Therefore, to make this document simple and concise the test cases within are limited to just one common input set and a few non common inputs. In this way the project's quality and risk are managed without dedicating too much of the team's resources to the testing phase of the project. Granted, if during testing the team feels that inputs not specified within this document are needed they may be added to the necessary test cases.

1.3. Conventions

To conform to industry standards and make the test cases within this document easily read and understood, the test cases have been written following a suggested template found within Bill Laboon's bool, "A Friendly Introduction to Software Testing." This template is outlined below along with a brief explanation of each component therein.

- **Test Case Name:** a unique, non numerical name to be used to refer to each case in discussions.
- **Identifier:** A unique identifier for each test case that is meant to be numerical in nature, but also contains a character abbreviation that identifies the purpose of the test case.

- **Description:** A <u>short</u> one to two sentence description of the test case.
- **Preconditions:** Any conditions that must be true before executing the test case. (Not applicable to every case) The abbreviation N/A is used if this section is not needed to define any preconditions of the test case.
- **Input Values:** Any input values to be passed in as part of the execution steps. If testing a set of input values, it should be specified in a mathematical format. (i.e. for the integer set 1 to 100 write 1,2,3,...,100)
- **Execution Steps:** A numbered list of steps the tester should take to run the test case.
- **Expected Output:** Any output values that should be expected for the test case to be successful.
- **Postconditions:** Any conditions that should be true after the test case has been run. The abbreviation N/A is used if this section is not needed to define any postconditions of the test case.

1.4. Intended Audience

The intended audience for this document are the team members of Project Thor that will be involved in the project's testing phase. However, this document may also be used by the faculty advisor for project evaluation purposes. Additionally, others who want to gain an understanding of how the project will be verified and validated may read this document to become more familiar with that process.

1.5. References

This document makes some references to external resources not defined or specified herein. Therefore, for the readers ease of reference they have been listed and linked below.
- Project Thor: Project Plan
- Project Thor: Requirements Document
- Project Thor: Design Document

2. Acceptance Testing

Acceptance testing is a test conducted to determine if the requirements of a specification are met. This section focuses on defining test cases that have that purpose

in mind. However, it should be noted that some of the test cases specified here may also help test use cases as well.

2.1. ToolTip Test

Identifier: Text here
Requirements: WEB-1, WEB-1.1
Description: Verifying behavior in accordance with requirements.
Preconditions: The user is accessing the web application using a web browser. Element has a ToolTip.
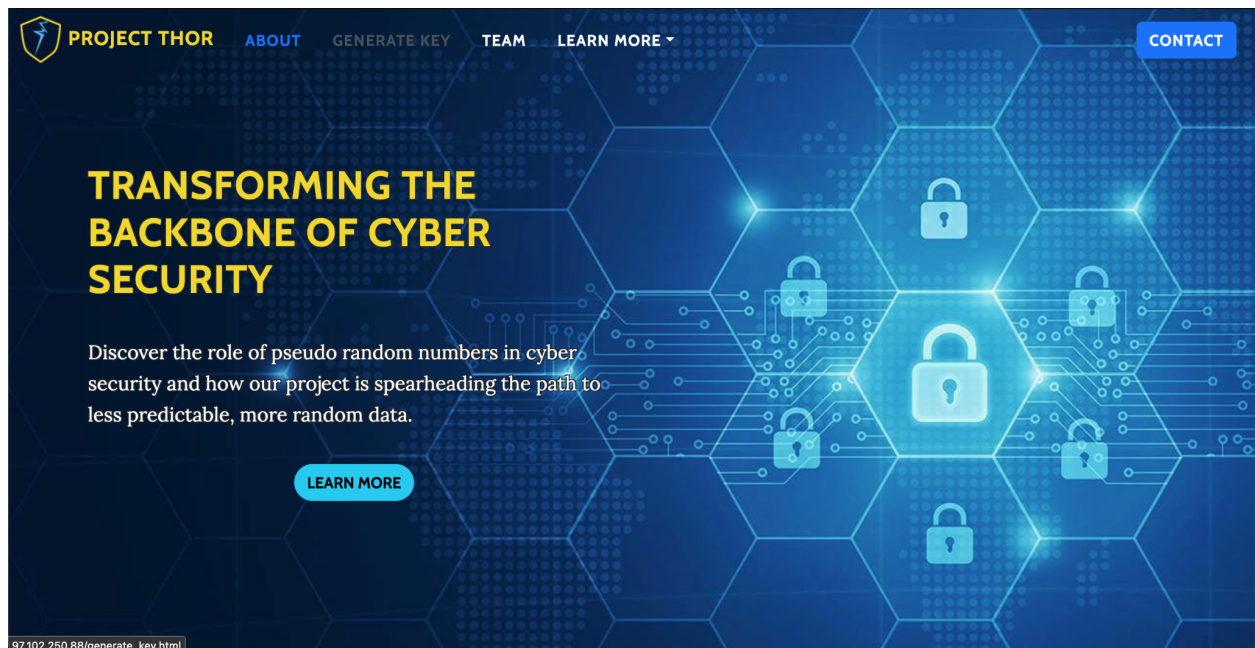**Input Values:** The cursor hovers on top of the element.
Execution Steps:

1. Navigate to the site
2. Hover over the "Generate Key" button

Expected Output: ToolTip popup displays within design requirements.
Postconditions: N/A.

RESULTS:
Failed: Mouse hovering over icons did not produce a ToolTip popup.

2.2. Generate Key Button Test

Identifier: Text Here.

Requirements: WEB-2, WEB 8.1

Description: Verifying key is presented when the Generate Key button is pressed.

Preconditions: The user is accessing the web application with a web browser.

Input Values: "Generate Key" button is pressed.

Execution Steps:
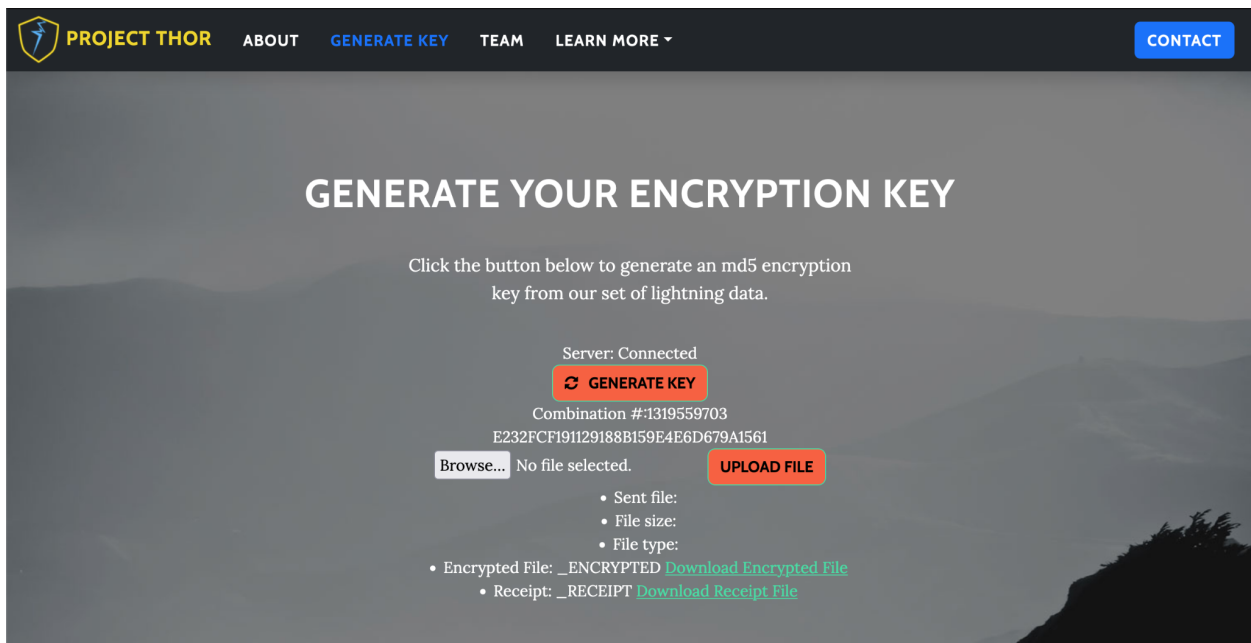
1. Navigate to site home page
2. Click the "Generate Key" Button

Expected Output: The random encryption key is displayed.

Postconditions: N/A.

RESULTS:

Passed: "Generate Key" button produces desired result.

2.3. Key Geography Location Test

Identifier: Text Here.
Requirements: WEB-3, WEB-3.1, WEB-12.3
Description: Verify the generated key is accompanied by data detailing the location of the associated lightning strike. The geographical data should be different when each key is generated.
Preconditions: The user pressed "Generate Key" and a key is displayed.
Input Values: N/A.
Execution Steps:

1. Generate Key button is pressed
2. Scroll down ½ a page
3. View geographical information in the Key Biography section

Expected Output: Location data lightning attributes are shown
Postconditions: N/A

RESULTS:
Passed: Geographical information is displayed.

2.4. Supported Browser Testing

Identifier: Text Here.
Requirements: WEB-4, WEB-5
Description: Verifying the browsers and operating systems listed in requirements correctly display the website.
Preconditions: Chrome, Safari, and Firefox are running Windows, macOS, iPadOS, iOS, Linux and Android systems.
Input Values: N/A
Execution Steps:

1. Navigate to website
2. Visit each of the 4 pages of the website

Expected Output: The website displays the same readable information in various screen sizes. Website will look the same on all PC systems, and on all mobile operating systems.
Postconditions: N/A

RESULTS:
Passed: All web browsers display the same.

2.5. Database Speed Test

Identifier: Text Here.

Requirements: WEB-6

Description: Verify website and database are fast.

Preconditions: The user is accessing the web application on a supported browser.

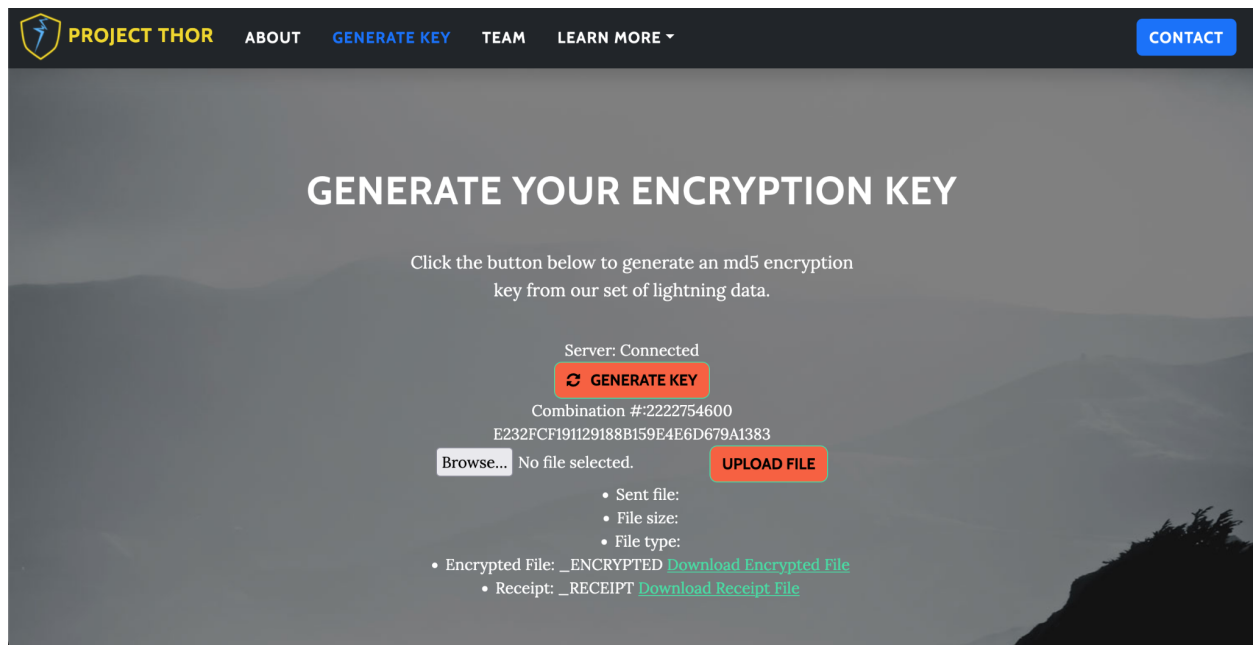**Input Values:** User presses generate key button

Execution Steps:

1. Navigate to website
2. Press the "Generate Key" button

Expected Output: Encryption key load and Key Biography appears within 3 seconds.

Postconditions: N/A

RESULTS:

Passed: Website reacts within 3 seconds.

2.6.   Website Speed Test

Identifier: Text Here.

Requirements: WEB-7, WEB-8.2, WEB-12.2

Description: The connection between the website and the server is verified to be fast.

Preconditions: The user is accessing the web application on a supported browser with an internet speed of at least 50mbps.

Input Values: Contact form button is submitted
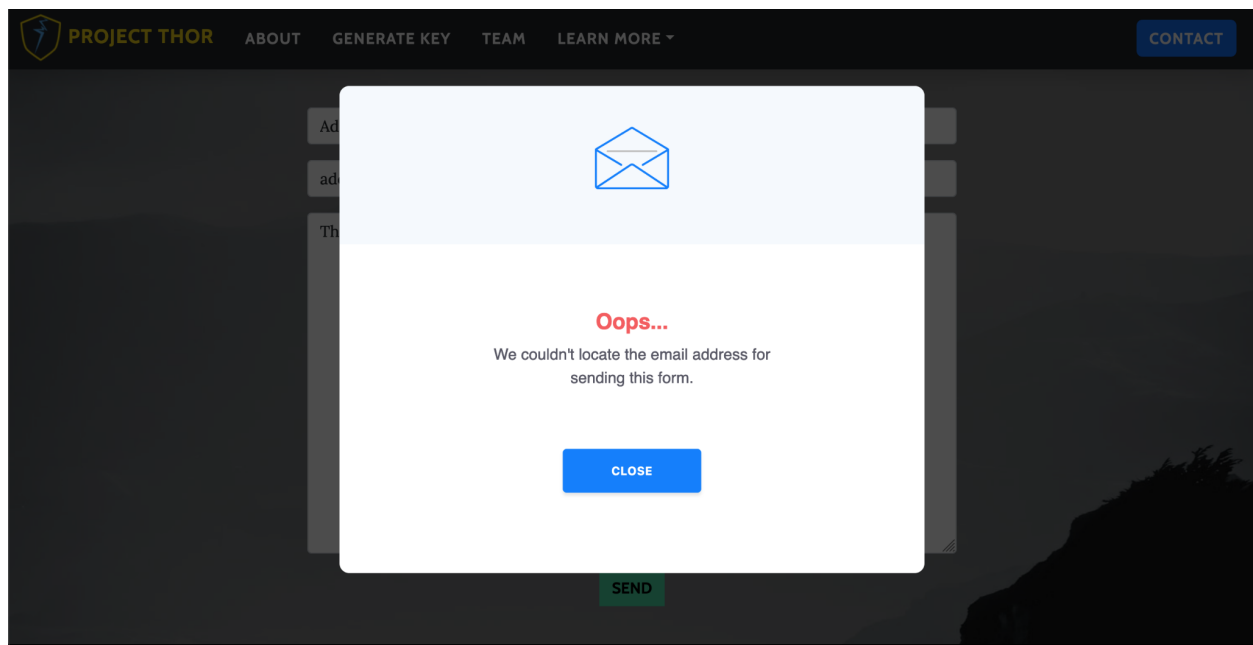
Execution Steps:

1. Navigate to site
2. Navigate to Contact Page
3. Fill out the contact form
4. Press submit button of the contact form

Expected Output: Form confirmation success message is received within 1 second.

Postconditions: Email is sent to the team within 1 second of form submission.

RESULTS:

Failed: We currently cannot email individual members.

2.7.   Availability Testing

Identifier: Text Here.

Requirements: DATA-1, WEB-9

Description: Verify availability of server database and website.

Preconditions: Third-party connection must be configured via Uptime Robot for free. This service will monitor website uptime every 5 minutes for free.

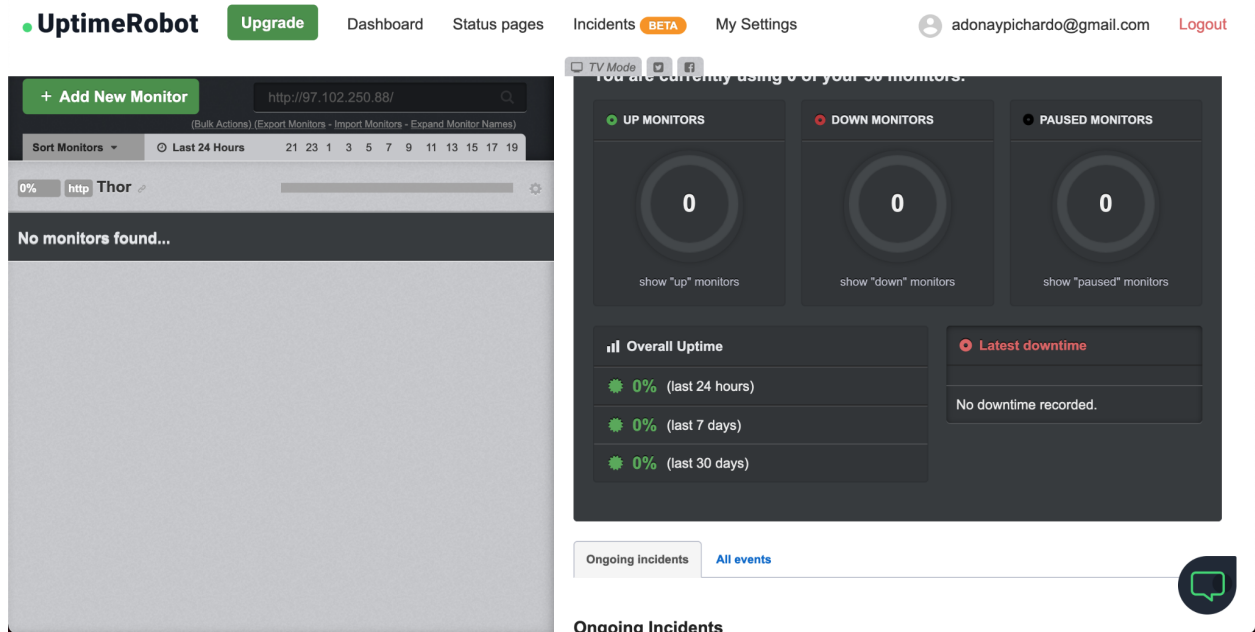**Input Values:** Website address.

Execution Steps:

1. Navigate to Uptimerobot.com
2. Enter website address to be tracked
3. View results

Expected Output: 100% uptime over 90 days

Postconditions: Operational or non-operational.

RESULTS:

On-Going: Not enough days have passed to consider this test passed, yet.

2.8. Database Maintenance

Identifier: Text Here.

Requirements: DATA-1, DATA-2, DATA-3

Description: Verify that the database is hosted on a team members machine that is possible to be connected to through SSH for maintenance and capable of rollback.

Preconditions: N/A

Execution Steps:

1. Verify that the database is hosted on the team members local machine
2. From a machine, hosted on an outside network, connect to the database machine via SSH
3. Verify that the database is capable of being rolled back from an existing version

Expected Output: The database shall be capable of being maintained form an outside source via SSH

Postconditions: Operational or non-operational

RESULTS:
PENDING

2.9. Database Real Time Update

Identifier: Text Here.

Requirements: DATA-6.1, DATA-8, DATA-11, DATA-12

Description: Verify that the database is capable of having real time data be added to the database from the ASCII data. As the data is being added, the database will confirm that there is no duplicate data as well as low-entropy data entered in the database.

Preconditions: Web Application should have a connection with the Database

Execution Steps:

1. As the ASCII data is updated through data gathering service, the database is updated as well
2. As the data is updated to the database, verify that no duplicate data is being entered in the database
3. As the data is updated to the database, verify that no low-entropy data is being entered in the database

Expected Output: As updated are made to the ASCII file, the database is being updated with high entropy and non-duplicate

RESULTS:
PENDING

2.10. SQL Request Testing

Identifier: Text Here

Requirements: DATA-4, DATA-6, DATA-10

Description: Verify the database's ability to receive sanitized requests from the web application.

Preconditions: Website connection with the Database

Input Values: Database address

Execution Steps:

1. Ensure the connection is made between the database and the website
2. From the web application, send the SQL request
3. As the SQL request is received, the request will be sanitized using standard sanitization techniques
4. Verify that data is received
5. Verify the data is current

Expected Output: The web application displays the most updated version of the data set requested from the database via the SQL request.

Postconditions: Operational or non-operational

RESULTS:
PENDING

3.   Use Case Testing

3.1.   Accessing Sup-pages Test

Identifier: Text Here
Requirements: WEB-8.2, WEB-12.3
Description: Verify that the user is able to access the home page, contact page, about page, and team page.
Preconditions: N/A
Input Values: N/A
Execution Steps:

1.Verify that upon the user entering the website URL, they are presented with team website homepage
2. Verify that the team member page has each users profile, photo, members name, major, professional about me, link to resume
3. Verify that the contact page has the contact information for the team
4.Verify that the about page has relevant information about the projects background and purpose

Expected Output: All pages are accessible to the user and they are able to access all expected information
Postconditions: N/A

RESULTS:
Failed:
1. Missing team member pictures
2. Missing emails
3. Missing Instagram links

PROJECT THOR

PROJECT THOR   ABOUT   GENERATE KEY   TEAM   LEARN MORE ▾   CONTACT

MEET THE TEAM

Our team of Computer Science and Software Engineering
Seniors draws on broad skills and experience to help make
your information more secure

JARED BLANCO
SOFTWARE ENGINEERING

Hello, I'm a Software Engineering at

LUKE BONENBERGER
COMPUTER SCIENCE

Software engineer with 6+ years of full

ADONAY PICHARDO
COMPUTER SCIENCE

Software Development/ Cyber Security

3.2. Data Explanation Test

Identifier: Text Here

Requirements: WEB-3

Description: Verify that the user is presented with a visual representation of how the data is generated

Preconditions: N/A

Input Values: N/A

Execution Steps:

1. Verify that upon loading the homepage of the web application, the user is presented with the visual representation of data,

2. Verify that the visual representation will include a live representation of the lightning from the location that data is being gathered from.

3. Verify that there will also be a map that shows a pin being dropped in a location of lighting, where sample data would be generated off the lightning strike

Expected Output: The user is presented with the graphical representation of data generation that contains enough explanation to create a shared understanding of the process.

Postconditions: N/A
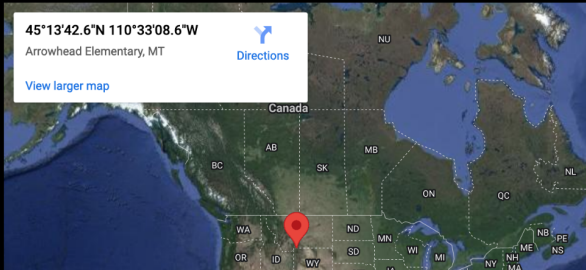
RESULTS:

Passed: Correct data is displayed.

KEY: E232FCF191129188B159E4E6D679A1561
HASH: CF31C6B990A0F77E8947A8321A049453
BIOGRAPHY

LOCATION

45.2285, -110.5524

45°13'42.6"N 110°33'08.6"W
Arrowhead Elementary, MT
Directions
View larger map

3.9 FALL — fall time

11.8 SECONDS — lifetime

-2 MWH

638722304 NANO SECS