



MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Website Checker System

Group 12 - IS	
Group members	Nguyễn Trương Thuý Vi – SE62116 (Leader) Lê Ngọc Trường – SE61777 Nguyễn Đức Trịnh – SE62136 Trần Phúc Anh – SE61789
Supervisor	Nguyễn Huy Hùng
Ext. Supervisor	N/A
Project Code	WCS

– Ho Chi Minh City, 12 September, 2018 –

Table of contents

A. <i>Introduction</i>	17
1. Project Information	17
2. Introduction.....	17
3. Current Situation	17
4. Problem Definition	18
5. Proposed Solution	18
5.1 Feature functions	18
5.2 Advantages and disadvantages	18
6. Functional Requirements.....	19
7. Role & Responsibility.....	19
B. <i>Software Project Management Plan</i>	20
1. Problem Definition	20
2. Project Organization.....	23
3. Project Management Plan	25
4. Coding Convention	26
C. <i>Software Requirement Specification</i>	26
1. User Requirement Specification	26
2. System Requirement Specification	27
3. Software System Attribute	88
4. Conceptual Diagram	89
D. <i>Software Design Description</i>	90
1. Design Overview.....	90
2. System Architectural Design	91
3. Component Diagram.....	92
4. Detailed Description	93
5. Interface	127
5.1 User Interface Design.....	127
5.1.1 Login	127

5.1.2	Menu Manager	128
5.1.3	Manage Staff.....	129
5.1.4	Manage Website.....	130
5.1.5	Menu Staff	131
5.1.6	Sitemap.....	133
5.1.7	Spelling test	134
5.1.8	Grammar test.....	135
5.1.9	Broken page test.....	136
5.1.10	Broken links test	137
5.1.11	Missing file test.....	138
5.1.12	Prohibited content test.....	139
5.1.13	Mobile layout test.....	140
5.1.14	Speed test.....	141
5.1.15	Javascript error test	143
5.1.16	Favicon test.....	144
5.1.17	Cookie test	145
5.1.18	Server behavior test.....	146
5.1.19	Contact detail test.....	147
5.1.20	Redrection test	148
5.1.21	Page test	149
6.	Database Design	150
6.1.	Entity relationship diagram (ERD).....	150
6.2.	Data Dictionary.....	151
7.	Algorithms	157
E.	<i>System Implementation & Test.....</i>	168
1.	Introduction.....	168
2.	Data Relationship Diagram	169
3.	Test Plan	171
3.1.	Features To Be Tested.....	171

3.2. Features Not To Be Tested	172
4. System Testing Test Case	172
<i>F. Software User's Manual.....</i>	187
<i>G. Appendix.....</i>	201

List of tables

Table 1 - Hardware Requirements for Server	22
Table 2 - Hardware requirement for PC	22
Table 3 - Software Requirement.....	23
Table 4 - Roles and responsibilities	24
Table 5 - Tools and techniques.....	25
Table 6 - Deliverables.....	25
Table 7 <Guest> Login use case specification	30
Table 8 - <Guest> Test broken links use case specification	32
Table 9 - <Guest> Test missing file use case specification.....	33
Table 10 - <Staff> Determine specific pages to test use case specification	36
Table 11 - <Staff> View Sitemap use case specification.....	37
Table 12 - <Staff> View pages referencing to the specific page use case specification.....	39
Table 13 - <Staff> Test spelling use case specification	41
Table 14 - <Staff> Test Grammar use case specification	42
Table 15 - <Staff> Test prohibited for website's content use case specification	44
Table 16 - <Staff> Test broken links use case specification	45
Table 17 - <Staff> Test broken pages use case specification	47
Table 18 - <Staff> Test missing file use case specification.....	49
Table 19-<Staff> Test the mobile layout use case specification.....	50
Table 20 - <Staff> Measure the speed of each page use case specification.....	52
Table 21 - <Staff> Test JavaScript errors use case specification.....	53
Table 22- <Staff> Test favicon use case specification	55
Table 23 - <Staff> Test cookies use case specification.....	56
Table 24 - <Staff> Test server behavior use case specification	58
Table 25 - <Staff> Test contact details on the website use case specification...59	59

Table 26 - <Staff> View all pages that contain the contact defined use case specification.....	61
Table 27 - <Staff> Test redirection use case specification.....	62
Table 28 - <Staff> Test pages use case specification.....	64
Table 29 - <Staff> Create report use case specification.....	65
Table 30 <Staff> View history reports use case specification.....	67
Table 31 - <Manager> Build sitemap use case specification.....	69
Table 32 - <Manager> View sitemap use case specification	71
Table 33 - <Manager> View pages referencing on the specific page use case specification.....	72
Table 34 - <Manager> Add new website use case specification	74
Table 35 - <Manager> Delete website use case specification.....	75
Table 36 - <Manager> Create new staff use case specification.....	77
Table 37 - <Manager> Delete staff use case specification	78
Table 38 <Manager> Assign staff use case specification	79
Table 39 <Manager> Add new word website's dictionary use case specification	81
Table 40<Manager> Delete word website's dictionary use case specification.	82
Table 41 <Manager> Add new warning word use case specification.....	84
Table 42 <Manager> Delete warning word use case specification.....	85
Table 43 <Manager> Edit warning word use case specification.....	87
Table 44 <Manager> View all reports use case specification	88
Table 45 Conceptual Diagram Data Dictionary	90
Table 46 Front End Component Diagram Dictionary	92
Table 47 Back End Component Diagram Dictionary.....	93
Table 48 Class Diagram Dictionary	96
Table 49<Class Diagram Attributes> Account.....	97
Table 50 <Class Diagram Methods> Account.....	97
Table 51 <Class Diagram Attributes> Role.....	97
Table 52<Class Diagram Methods> Role	97

Table 53 <Class Diagram Attributes> Website	98
Table 54 <Class Diagram Methods> Website	98
Table 55 <Class Diagram Attributes> Dictionary	98
Table 56 <Class Diagram Methods> Dictionary	98
Table 57 <Class Diagram Attributes> Sitemap	99
Table 58 <Class Diagram Methods> Sitemap	99
Table 59 <Class Diagram Attributes> Warningword	99
Table 60 <Class Diagram Methods> Warningword	99
Table 61 <Class Diagram Attributes> Topic	100
Table 62 <Class Diagram Methods> Topic	100
Table 63 <Class Diagram Attributes> Page	100
Table 64 <Class Diagram Methods> Page	101
Table 65 <Class Diagram Attributes> Version	101
Table 66 <Class Diagram Attributes> GrammarReport	102
Table 67 <Class Diagram Attributes> SpellingReport	102
Table 68 <Class Diagram Methods> SpellingReport	102
Table 69 <Class Diagram Attributes> WarningWordReport	103
Table 70 <Class Diagram Attributes> PageReport	103
Table 71 <Class Diagram Attributes> ContactReport	104
Table 72 <Class Diagram Attributes> BrokenLinkReport	104
Table 73 <Class Diagram Attributes> RedirectionReport	104
Table 74 <Class Diagram Attributes> BrokenPageReport	105
Table 75 <Class Diagram Attributes> MissingFileReport	105
Table 76 <Class Diagram Attributes> FaviconReport	106
Table 77 <Class Diagram Attributes> SpeedReport	106
Table 78 <Class Diagram Attributes> JavascriptReport	106
Table 79 <Class Diagram Attributes> MobileLayoutReport	107
Table 80 <Class Diagram Attributes> CookieReport	107
Table 81 <Class Diagram Attributes> ServerBehaviorReport	108

Table 82 <Fields> Login.....	127
Table 83 <Buttons/Hyperlinks> Login.....	127
Table 84 <Buttons/Hyperlinks> Menu Manage.....	128
Table 85 <Fields> Manage Staff	129
Table 86 <Buttons/Hyperlinks> Manage Staff	129
Table 87 <Fields> Manage Website.....	130
Table 88 <Buttons/Hyperlinks> Manage Website.....	131
Table 89 <Buttons/Hyperlinks> Menu Staff.....	133
Table 90 <Fields> Sitemap	134
Table 91 <Buttons/Hyperlinks> Sitemap	134
Table 92 <Fields> Spelling Test	134
Table 93 <Buttons/Hyperlinks> Grammar Test.....	135
Table 94 <Fields> Grammar Test.....	135
Table 95 <Buttons/Hyperlinks> Grammar Test.....	136
Table 96 <Fields> Broken Pages Test	136
Table 97 <Buttons/Hyperlinks> Broken Pages Test	137
Table 98 <Fields> Broken Links Test.....	137
Table 99 <Buttons/Hyperlinks> Broken Links Test.....	138
Table 100 <Fields> Missing File Test	139
Table 101 <Buttons/Hyperlinks> Missing File Test	139
Table 102 <Fields> Prohibited Content Test.....	140
Table 103 <Buttons/Hyperlinks> Prohibited Content Test.....	140
Table 104 <Fields> Mobile Layout Test.....	141
Table 105 <Buttons/Hyperlinks> Mobile Layout Test	141
Table 106 <Fields> Speed Test.....	142
Table 107 <Buttons/Hyperlinks> Speed Test.....	142
Table 108 <Fields> JavaScript Errors Test	143
Table 109 <Buttons/Hyperlinks> JavaScript Errors Test	143
Table 110 <Fields> Favicon Test.....	144

Table 111 <Buttons/Hyperlinks> Favicon Test.....	145
Table 112 <Fields> Cookie Test.....	145
Table 113 <Buttons/Hyperlinks> Cookie Test.....	146
Table 114 <Fields> Server Behavior.....	146
Table 115 <Buttons/Hyperlinks> Server Behavior.....	146
Table 116 <Fields> Contact Details.....	147
Table 117 <Buttons/Hyperlinks> Contact Details Test	148
Table 118 <Fields> Redirection Test.....	148
Table 119 <Buttons/Hyperlinks> Redirection Test.....	148
Table 120 <Fields> Page Test	149
Table 121 <Buttons/Hyperlinks> Page Test	149
Table 122 Entity Relationship Diagram Dictionary	152
Table 123 Physical Diagram Dictionary.....	171
Table 124 <Test case> Manage Page Option.....	173
Table 125<Test case> Broken Links Test.....	174
Table 126 <Test case> Broken Pages Test.....	175
Table 127 <Test case> Prohibited Content Test.....	176
Table 128 <Test case> Mobile Layout Test	176
Table 129 <Test case> Speed Test.....	177
Table 130 <Test case> Javascript Errors Test.....	178
Table 131 <Test case> Cookie Law Test.....	179
Table 132 <Test case> Server Behavior Test	180
Table 133 <Test case> Missing File Test	181
Table 134 <Test case> Favicon Test	182
Table 135 <Test case> Contact Details Test.....	183
Table 136 <Test case> Redirection Test.....	183
Table 137 <Test case> Pages Test.....	184
Table 138 <Test case> Spelling Test.....	185
Table 139 <Test case> Grammar Test	186

Table 140 - Hardware requirement for server.....	187
Table 141 - Software requirement for server	188
Table 142 - Add Staff Steps.....	190
Table 143 - Edit Staff Steps.....	190
Table 144 - Delete Staff Steps.....	191
Table 145 - Make new version Steps	192
Table 146 - View Sitemap Steps.....	192
Table 147 - Assign website steps	193
Table 148 - Add warning word steps	194
Table 149 - Edit warning word steps	196
Table 150 - Select website steps.....	196
Table 151 - Add page option steps	198
Table 152 - Rename or Delete page option steps	199
Table 153 - Do a test function steps.....	200

List of Figures

Figure 1 - Scrum Framework.....	23
Figure 2 - System Overview Use Case	28
Figure 3 - Guest Overview Use Case.....	29
Figure 4 - <Guest> Login.....	29
Figure 5 - <Guest> Test broken links.....	30
Figure 6 - <Guest> Test missing file.....	32
Figure 7 - Staff Overview Use Case	34
Figure 8-<Staff> Determine specific pages to test	34
Figure 9 - <Staff> View sitemap	36
Figure 10 - <Staff> View pages referencing to the specific page.....	38
Figure 11 - <Staff> Test spelling	39
Figure 12 - <Staff> Test grammar	41
Figure 13 - <Staff> Test prohibited for website's content	42
Figure 14 - <Staff> Test broken links	44
Figure 15 - <Staff> Test broken pages	46
Figure 16 - <Staff> Test missing file.....	47
Figure 17 - <Staff> Test the mobile layout.....	49
Figure 18 - <Staff> Measurethe speed test of each page.....	51
Figure 19 - <Staff> Test JavaScript errors.....	52
Figure 20-<Staff> Test favicon.....	53
Figure 21 - <Staff> Test cookies.....	55
Figure 22 - <Staff> Test server behavior	57
Figure 23 - <Staff> Test contact details on the website.....	58
Figure 24 - <Staff> View all pages that contain the contact defined	60

Figure 25 - <Staff> Test redirection.....	61
Figure 26 - <Staff> Test pages.....	63
Figure 27 <Staff> Create report	64
Figure 28 - <Staff> View history reports	65
Figure 29 - Manager Overview Use Case	67
Figure 30 - <Manager> Build Sitemap	68
Figure 31 - <Manager> View sitemap	70
Figure 32 - <Manager> View pages referencing on the specific page	71
Figure 33 - <Manager> Add new website.....	73
Figure 34 <Manager> Delete website	74
Figure 35 - <Manager> Create new staff.....	75
Figure 36 - <Manager> Delete staff.....	77
Figure 37 <Manager> Assign staff	78
Figure 38 <Manager> Add new word website's dictionary	80
Figure 39 <Manager> Delete word website's dictionary.....	81
Figure 40 <Manager> Add new warning word.....	83
Figure 41 <Manager> Delete warning word.....	84
Figure 42 <Manager> Edit warning word.....	86
Figure 43 <Manager> View all reports.....	87
Figure 44 Conceptual Diagram.....	89
Figure 45 Software Architecture Design.....	91
Figure 46 Front End Component Diagram.....	92
Figure 47 Back End Component Diagram	93
Figure 48 Class Diagram	94
Figure 49 <Sequence Diagram> Guest Login.....	109
Figure 50 <Sequence Diagram> Staff Test Pages.....	110
Figure 51 <Sequence Diagram> Staff Test Favicon	111
Figure 52<Sequence Diagram> Staff Test Contact Details.....	112
Figure 53 <Sequence Diagram> Staff Test Redirection.....	113

Figure 54 <Sequence Diagram> Staff Test Missing File.....	115
Figure 55 <Sequence Diagram> Manager Build Sitemap	116
Figure 56 <Sequence Diagram> Staff Test Cookie	117
Figure 57 <Sequence Diagram> Staff Test Broken Pages.....	118
Figure 58 <Sequence Diagram> Staff Test Broken Links.....	119
Figure 59 <Sequence Diagram> Staff Test Mobile Layout	120
Figure 60 <Sequence Diagram> Staff Test Javascript Errors.....	121
Figure 61 <Sequence Diagram> Staff Test Prohibited	122
Figure 62 <Sequence Diagram> Staff Test Spelling.....	123
Figure 63 <Sequence Diagram> Staff Test Grammar	124
Figure 64 <Sequence Diagram> Staff Test Server Behavior	125
Figure 65 <Sequence Diagram> Staff measure the speed test.....	126
Figure 66 Login	127
Figure 67 Menu Manager	128
Figure 68 Manage Staff.....	129
Figure 69 Manage Website	130
Figure 70 Menu Staff	131
Figure 71 Sitemap.....	133
Figure 72 Spelling Test.....	134
Figure 73 Grammar Test.....	135
Figure 74 Broken Pages Test.....	136
Figure 75 Broken Links Test.....	137
Figure 76 Missing File Test.....	138
Figure 77 Prohibited content test.....	139
Figure 78 Mobile Layout Test	140
Figure 79 Speed Test.....	141
Figure 80 - JavaScript Errors Test.....	143
Figure 81 Favicon Test	144
Figure 82 Cookie Test	145

Figure 83 Server Behavior.....	146
Figure 84 - Contact Details Test.....	147
Figure 85 - Redirection Test.....	148
Figure 86 Page Test.....	149
Figure 87 Entity Relationship Diagram.....	150
Figure 88 Define Sitemap.....	157
Figure 89 Trie Tree	162
Figure 90 Structure Trie Tree	163
Figure 91 Physical Diagram	169
Figure 92 - Add Staff UI	189
Figure 93 - Add Staff Popup	189
Figure 94 - Edit Staff UI	190
Figure 95 - Edit Staff Popup	190
Figure 96 - Delete Staff UI	191
Figure 97 - Delete Staff Popup	191
Figure 98 - Make new version UI.....	191
Figure 99 - View Sitemap UI	192
Figure 100 - Sitemap Screen.....	192
Figure 101 - Assign website UI	193
Figure 102 - Assign website popup	193
Figure 103 - Assign website popup	193
Figure 104 - Add warning word UI.....	194
Figure 105 - Add warning word popup.....	194
Figure 106 - Add warning word popup.....	194
Figure 107 - Edit warning word UI.....	195
Figure 108 - Edit warning word popup.....	195
Figure 109 - Edit warning word popup.....	195
Figure 110 - Select website UI	196
Figure 111 - Select page option UI.....	196

Figure 112 - Add page option popup.....	197
Figure 113 - Add page option popup.....	197
Figure 114 - Add page option popup.....	197
Figure 115 - Select page option UI.....	198
Figure 116 - Select page option popup.....	198
Figure 117 - Rename or Delete page option popup.....	199
Figure 118 - Select page option UI.....	199
Figure 119 - Select page option popup.....	200
Figure 120 - Test function UI.....	200

A. Introduction

1. Project Information

- Project name: **Website Checker System**
- Project Code: **WCS**
- Product Type: **Website Application**
- Start Date: **September 12nd, 2018**
- End Date: --

2. Introduction

Nowadays, the website design and development is rising rapidly. Businesses use the website as a main channel for providing information, products, and services to partners and potential customers. A good website does not only have a good look but also needs to operate correctly. So, web development is so important to every company, even brand-new startups. If only concerned with the design of a website and not the functionality – a business won't be as successful at attracting new users to their website because they'll be lacking in the area of user experience.

In this document, we introduce a solution for web owners: a web application called Website Checker System. The Website Checker System will be a useful assistance to enterprise ensuring the quality of the information their website provides, how easily their users can navigate the site, and how functional their website is. This document also describes our working process in 4 months including our perspective on the system, component designs and detailed core workflows.

3. Current Situation

The higher a website's site quantity, the more complicated the test process is. Assuming entire processes are manual operations, it's certainly costly, time-consuming. In many cases, there are underlying issues: JavaScript errors, bad server behavior, responsive on multi devices.

There are many reasons which cause negative effects on operational business. Slow speed, low user experience, or risks such as broken links, missing files, etc. Also, the number of potential customers of enterprises may decrease when their website does not navigate well. Mistakes in spelling, grammar also directly influence the enterprise's trademark.

4. Problem Definition

Below are the disadvantages of the current situation:

- The Website owner can't ensure that entire their website run smoothly or function correctly as they expected.
- Broken links, broken pages or missing files can't be immediately detected.
- Some resources in website content are not up to date or no valid anymore.
- Technical properties: server behavior, page redirection are not clearly displayed. So that, Website owner not able to track the quality of their website in details.
- Even in critical information, spelling and grammar mistakes still exist.

Above problems may reduce the effectiveness of product sale, business ads, business's trademark. A poor quality website reflects on an entire brand.

5. Proposed Solution

Building a web application known as an automatic test system. It helps website owner tracking the entire process day by day. The following workflow and features should be implemented:

5.1 Feature functions

- Detect quality issues like spelling and grammar mistakes, missing files and broken links.
- Measure and improve the user experience of enterprise's website. Check that pages are whether fast, mobile-friendly, and work correctly or not.
- Ensure compliance against enterprise's own standards, including brand and legal.
- Discover what content, technology, and resources enterprise's websites use.

5.2 Advantages and disadvantages

The advantages and disadvantages of the proposed solution:

❖ Advantages:

- Test many things in one place.
- The system will promptly alert when something is in the warning state.
- Monitor website at any time, with explicit information on visual dashboard. Measure and improve the quality of website.

❖ Disadvantages:

- Only support for website in English.

6. Functional Requirements

Function requirements of the system are listed as below:

6.1 Manager:

- Manager can create, activate, de-activate or update account.
- Manager can manipulate on each repository: website's dictionary, warning word list.
- Manager can build, update website's sitemap with different versions.

6.2 Staff:

- User can view website's sitemap and check references of each page.
- User can check spelling and grammar mistakes.
- User can detect the broken element on website: broken links, broken pages.
- User can detect missing files in each page.
- User can scan warning words and check where they appear on each page.
- User can check mobile layout.
- User can know speed of website: how long to respond, when page become usable, when load completed.
- User can detect JavaScript errors on each page.
- User can check the favicons website using: missing, invalid, conformity.
- User can know which pages that collect user's cookie & purpose of using.
- User can track the server behavior.
- User can view all of page details, including URL path and its HTTP response code.
- User can list all of the contact details such as email, phone existing on the website. User can know each defined contact information appearing on which pages.
- User can test HTTP Redirection codes.
- User can export data to reports, and can review again by listing all history reports.

7. Role & Responsibility

No	Full Name	Role	Position	Contact
1	Nguyễn Huy Hùng	Project Manager	Supervisor	hungnh@fpt.edu.vn
2	Nguyễn Trương Thúy Vi	Developer	Leader	vinttse62116@fpt.edu.vn
3	Lê Ngọc Trường	Developer	Member	truonglnse61777@fpt.edu.vn
4	Nguyễn Đức Trinh	Developer	Member	trinhndse62136@fpt.edu.vn
5	Trần Phúc Anh	Developer	Member	anhtpse61789@fpt.edu.vn

B. Software Project Management Plan

1. Problem Definition

1.1 Name of this Capstone Project

- Official name: Website Checker System
- Vietnamese name: Hệ thống kiểm tra website
- Abbreviation: WCS

1.2 Problem Abstract

- This project is concern about the automation of a massive range of checks that would otherwise take an extraordinary amount of time and expertise to perform. We call it Website Checker System (WCS). WCS will provide a user-friendly interface to check and view result about enterprises' website.
- Businesses use the website as a main channel for providing information, products, and services to partners and potential customers. With the current checker system, enterprises only able to check performance on their website. Therefore, it's hard to evaluate the true quality of their website in other facets.
- The Website Checker System will be a useful assistance to enterprise ensuring the quality of the information their website provides, how easily their users can navigate the site, and how functional their website is.
- Finally, WCS gives result and present as charts, helps the enterprise to review their website.

1.3 Project Overview

1.3.1 Current Situation

Below are the problems encountered in this project:

- Limit time and human resource: Team has only 4 member and time for all project is about 13 weeks for writing a document, implementing the products and testing.
 - Schedule of team member: team members can have a conflict in meeting schedule because of sick, or class and work schedule, etc.
 - Framework study: team members have a problem when applying the play framework into project. The team needs an amount of time to get familiar new techniques.
 - New technique: Some team members are new to the techniques used in the project. The team needs an amount of time to get familiar with those techniques.
 - Lack of UI (user interface), UX (user experience) design skill: Our team members all study IS major, and no one has studied UI, UX design. Therefore, that some UI may misunderstand or hard to use with normal users.

1.3.2 The Proposed System

Building a web application that allows end users test their website. End users includes guests, managers and staffs. Managers can manage staff permission, manipulate on repositories. Staffs can test their website about quality, experience, technology and content on their website.

Task will be assigned vertically to team members, so that if one member quits, the team will not lack of resources.

1.3.3 Boundaries of the System

The system can:

- Allow Manager to manage user accounts .
- Allow Manager to manage dictionaries, warning words list.
- Allow Staff to test websites with specific features.
- Allow Staff to export report.
- Allow Staff to view the position of mistakes on the website, or decide to do with the result in some test case.

The system cannot:

- System just support Web Owner to test English websites.

1.3.4 Future Plans

❖ Current system can only export individually for each single function. We design the system to provide options for users:

- Export all test report: system will export all result test.
- Implement visualization of result test by charts.

❖ Current systems can only perform work under the influence of the user. We design the system that allows scheduling for automatic testing:

- Check by hour, day, month: The system will be scheduled to automatically test.
- Customizable functions for automatic testing: The system will allow user select each function with defined input parameters.

❖ Current system can only provide reports on the test results independently. We design the system to analysis and evaluation based on test results of related reports.

1.3.5 Development environment

1.3.5.1 Hardware requirements

For server

Hardware	Minimum Requirement	Recommended
Internet Connection	Cable, Wi-Fi (8 Mbps)	Cable, Wi-Fi (50 Mbps or more)

Operation System	XP, Vista, 7, 10, Window Server 2008	10, Window server 2008
Computer Processor	Intel® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer memory	4GB RAM	32 GB RAM or more
Storage space	1GB	5GB or more

Table 1 - Hardware Requirements for Server

For PC

PC	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Window 7	Window 7 or more.
Computer Processor	Intel® Core i3 1.4GHz	Intel® Core i5 2.50GHz
Computer Memory	1GB RAM	2GB RAM or more
Web Browser	Chromes (v42 or higher)	Chrome latest stable version

Table 2 - Hardware requirement for PC

1.3.5.2 Software requirement

Software	Name / Version	Description
Operation System	Windows Server 2014	Operating system and platform for development
Environment	Java EE 8.0, Node v10, npm v6	Specification for developing web application
Modeling tool	StarUML	Used to design diagram
IDE	Intellij IDEA 2018.1.5, Visual Studio Code 1.27.2, A5M2 2.12.1 (SQL)	Programming tools
DBMS	MYSQL 8.0, Elastic Search	Used to create & manage the database for system

Source control	Git on IDE (Git lab)	Used for source control
Web browser	Chrome 69 or above	Testing browser

Table 3 - Software Requirement

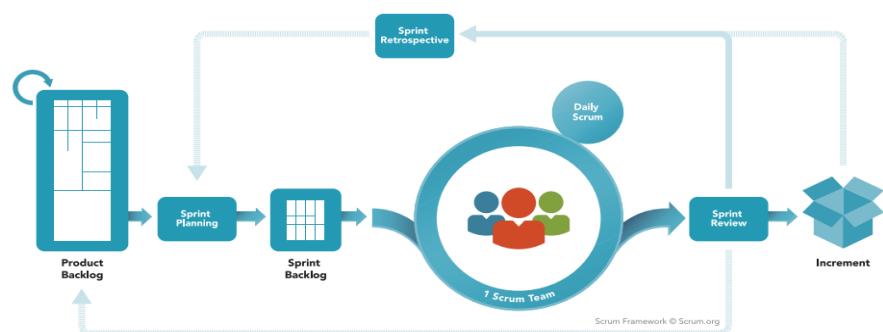
2. Project Organization

2.1 Software Process Model

This project is developed using Scrum model – part of an agile framework for Software development project. Our team choose Scrum model because of the following reasons:

- Our team only has 4 members, and tasks are assigned vertically, do all steps from design, coding, testing and implementation. Scrum is the most suitable model for small and medium project.
- After research about Trie tree, Radix tree data structure, Elastic search, we have defined problems. The risk of changing algorithm is high because proving accuracy of those algorithms is complicated. We need to use “try and test” method.
- The project contains complicated system and the concept of machine learning is very new for us, so we need to try many designs before the system run stability.
- In the project there are many new technologies that need to be learned. With the Scrum model, the team can learn and develop in parallel to meet deadline.
- There is no leader, no hierarchy in team, so team members work cheerfully, stimulating the initiative and creativity of each member.
- Product owner can change requirement or extend scope. The team will adapt to change better.

SCRUM FRAMEWORK



 Scrum.org

Figure 1 - Scrum Framework

2.2 Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Nguyễn Huy Hùng	Product Owner	<ul style="list-style-type: none"> • Specify user requirement • Control the development process • Give out technique and business analysis support
2	Nguyễn Trương Thúy Vi	Scrum Master	<ul style="list-style-type: none"> - Managing process - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing - Arrange Meeting - Risk Management
3	Lê Ngọc Trường	Scrum team member	<ul style="list-style-type: none"> - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing
4	Nguyễn Đức Trịnh	Scrum team member	<ul style="list-style-type: none"> - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing
5	Trần Phúc Anh	Scrum team member	<ul style="list-style-type: none"> - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing

Table 4 - Roles and responsibilities

2.3 Tools and Techniques

Tool/Technique	Name
Front-end	HTML, CSS, JavaScript, jQuery, React Semantic UI
Back-end	JavaEE, SpringBoot framework, JPA, Hibernate
IDE	NetBeans 8.2, IntelliJ IDEA 17.2
DBMS	MySQL
Source Control	GitLab
Modelling tool	StarUML

Table 5 - Tools and techniques

3. Project Management Plan

3.1. Product Backlog

Product Backlog could be found [here](#).

3.2. Sprint Backlog

Sprint Backlog can be found [here](#)

3.3. Deliverables

No	Deliverable	Note
1	Introduction, Entity Relationship Diagram, Use Case Overview, Mock UI	Sprint 1
2	Study Spring Boot Framework, React JS, Design User Interface for Web Application	Sprint 2
...		

Table 6 - Deliverables

3.4. All Meeting Minutes

All sprint meeting minutes could be found [here](#).

4. Coding Convention

Summary:

- **Naming Convention:**
 - Variable names should be short yet meaningful. The choice of a variable name should be designed to indicate to the casual observer the intent of its use.
 - Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.
- **Indentation:**
 - One declaration per line is recommended since it encourages commenting.
 - In absolutely no case should variables and functions be declared on the same line.
 - Do not put different types on the same line.
- **Declarations Convention:**
 - One declaration per line is recommended since it encourages commenting.
 - Using Java Code Convention from:

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

C. Software Requirement Specification

1. User Requirement Specification

1.1 Guest Requirement

Guest is a person who doesn't have access to the system. Guest only try some demo features. To use all functions, guest must login. These are some functions guest can use:

- Login.
- Test broken links.
- Test missing files.

1.2 Staff Requirement

Staff is an authorized user who has permission to login into the system. With staff role, user can use below functions:

- Determine specific pages to test.
- View sitemap.
- View pages referencing to the specific page.
- Test spelling for website's content.
- Test grammar for website's content.
- Test prohibited for website's content.
- Test broken links.

- Test broken pages.
- Test missing files.
- Test the mobile layout.
- Measure the speed test.
- Test JavaScript errors.
- Test Favicon.
- Test cookie website using.
- Test server behavior.
- Test contact details on the website.
- Test redirections.
- Test website's pages.
- Create report.
- View history reports.

1.3 Manager Requirement

Manager is an authorized user has permission to login into the system under Manager role. Manager can use below management functions:

- Build Sitemap.
- View sitemap.
- View pages referencing to the specific page.
- Manage staff permissions.
- Manage websites.
- Manage the website's dictionary.
- Manage the warning word list.
- View all reports.

2. System Requirement Specification

2.1 External Interface Requirement

2.1.1 User Interface

- GUI should be simple, clear, intuitive, and reminiscent.
- The interface design is an iterate process includes: design, sketching, prototyping, user assessment.
- The User Interface uses language is English.

2.1.2 Hardware Interface

- Desktop PC.

2.1.3 Software Interface

- Web application: work with browsers Firefox (v52 or higher), Chromes (v28 or higher), Internet Explorer (v10 or above) or with any web browser that supports HTML5 & CSS3.

2.1.4 Communication Protocol

Use HTTP protocol 1.1 for communication between:

Web application and web server

2.2 System Overview Use Case

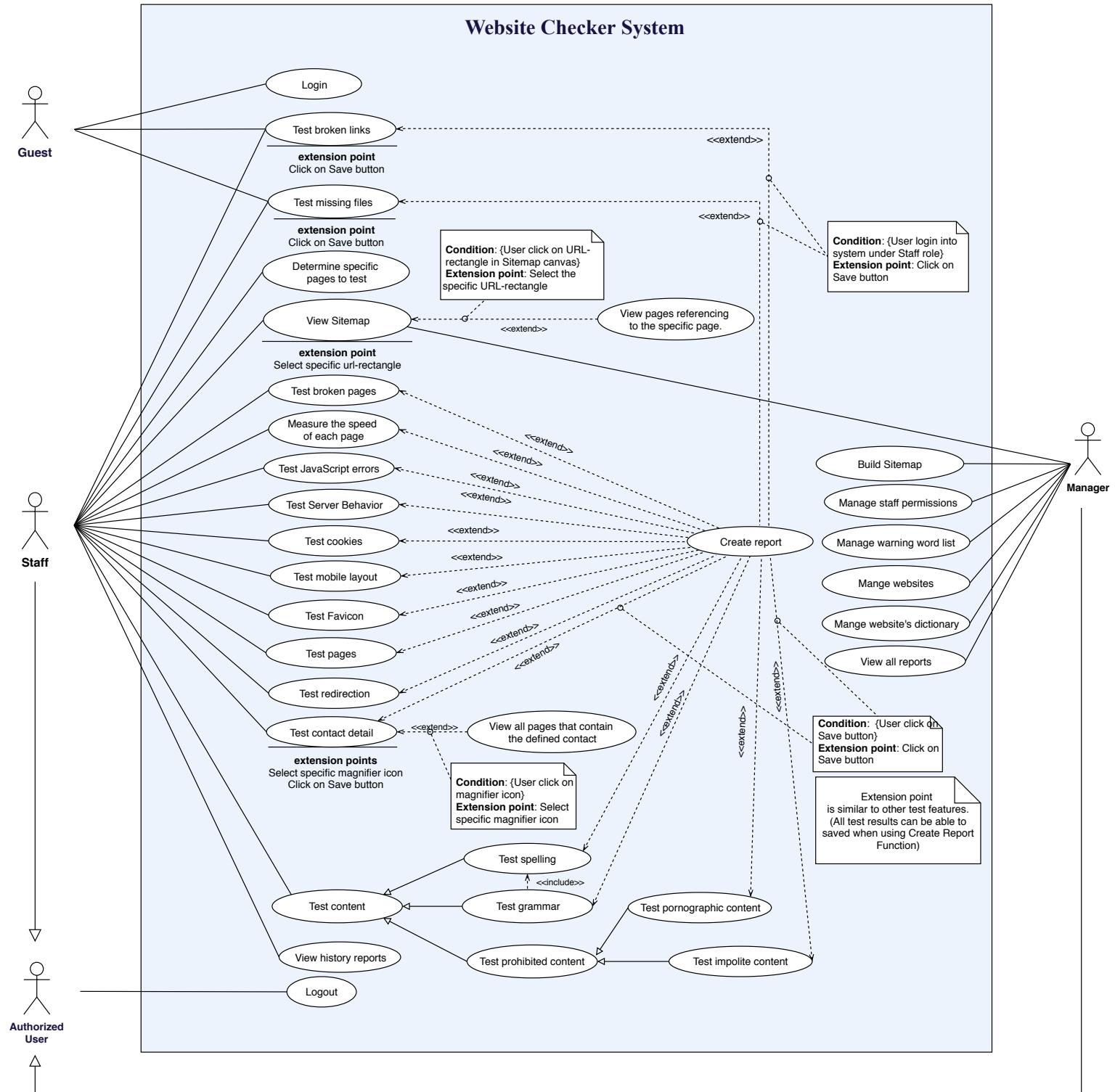


Figure 2 - System Overview Use Case

2.3 List of use case

2.3.1 <Guest> Overview Use Case

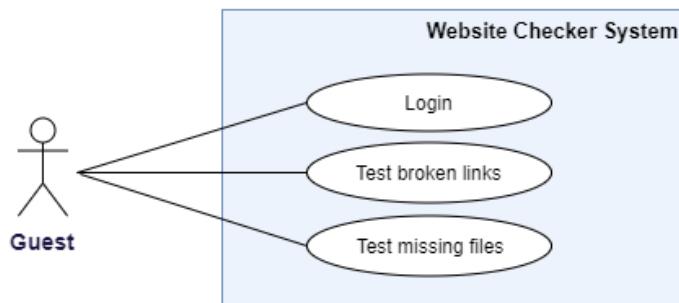


Figure 3 - Guest Overview Use Case

2.3.1.1 Login



Figure 4 - <Guest> Login

USE CASE - WCS_UC_1			
Use Case No.	WCS_UC_1	Use Case Version	1.0
Use Case Name	Login		
Author	ViNTT		
Date	24/10/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Guest 		
Summary:	<ul style="list-style-type: none"> - Allow guest to log in the system. 		
Goal:	<ul style="list-style-type: none"> - Guest can log in the system with specific role. 		
Triggers:	<ul style="list-style-type: none"> - Guest sends login command. 		
Preconditions:	N/A		
Post conditions:	<ul style="list-style-type: none"> - Success: Guest is exactly authorized with specific role in the system. - Fail: System shows error messages. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Guest goes to login view.	System requires identity information from Guest:	

		- Email or customer code: text input. - Password: text input.
2	Guest inputs information.	
3	Guest sends command to login to system.	Guest logsins system with his specific role. [Alternative 1] [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	Guest enter wrong identity information.	System shows error message "Wrong username or password. Please try again."

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection" when the internet is lost.

Relationships: N/A**Business Rules:**

- Password is encrypted before being sent to server.
- After login to system, guest will be redirected to specific view based on their role on the system: staff or manager.
 - If role is "Staff", the system will display to Staff view.
 - If role is "Manager", the system will display to Manager Dashboard view.

Table 7 <Guest> Login use case specification

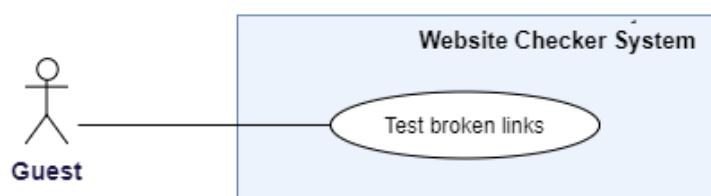
2.3.1.2 Test broken links

Figure 5 - <Guest> Test broken links

USE CASE - WCS_UC_2			
Use Case No.	WCS_UC_2	Use Case Version	1.0
Use Case Name	Test broken links		
Author	ViNTT		
Date	24/10/2018	Priority	Normal

Actor:

- Guest

Summary:

- This use case allow guest to detect broken links with the defined website's address.

Goal:

- Guest can view broken link's information: URL, HTTP status code and response message.

Triggers:

- Guest sends test broken links command.

Preconditions: N/A**Post conditions:**

- Success: Display the result test on screen.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest enter the address of a website at input field.	
2	Guest send command to test broken links.	System response the result with information: "Page", "Broken Link", "Issue". [Alternative 1] [Exception 1, 2]

Alternative Scenario:

No	Actor Action	System Response
1	Guest click out of the loading dialog.	<p>Display a dialog with message "Do you want to cancel Test Broken Links process?"</p> <ul style="list-style-type: none"> - "Yes" - system end the process and redirect guest back to landing page. - "No" – Continue to find broken links.

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection" when the internet is lost.
2	Guest enter invalid URL.	System shows message "400 Bad request"

Relationships: N/A**Business Rules:**

- Both internal and external links of the website are tested.
- Reasons that make links become broken are shown in detail.

- With guest role, the limit of page test is set. System only tests five pages of the input website's address. For more pages, guest must register account and login to use full other features.

Table 8 - <Guest> Test broken links use case specification

2.3.1.3 Test missing files

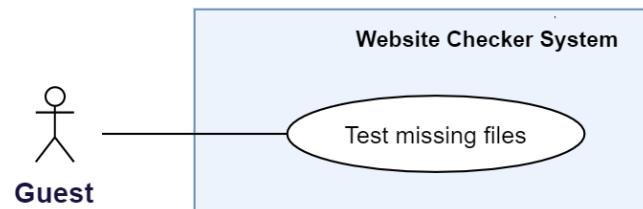


Figure 6 - <Guest> Test missing file

USE CASE - WCS_UC_3			
Use Case No.	WCS_UC_3	Use Case Version	1.0
Use Case Name	Test missing files		
Author	ViNTT		
Date	24/10/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Guest 		
Summary:	<ul style="list-style-type: none"> - This use case allows guest to detect missing file with the defined website's address. 		
Goal:	<ul style="list-style-type: none"> - Guest can view missing file information: file URL source, description missing of each page on the result screen. 		
Triggers:	<ul style="list-style-type: none"> - Guest sends test missing file command. 		
Preconditions: N/A			
Post conditions:	<ul style="list-style-type: none"> - Success: Display the result test on screen. - Fail: System shows error messages. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Guest enter the address of a website at input field.		
2	Guest send command to test missing files.	System response the result with information: "File", "Description", "Page affected". [Alternative 1] [Exception 1, 2]	
Alternative Scenario:			
No	Actor Action	System Response	
1	Guest click out of the loading dialog.	Display a dialog with message "Do you want to cancel Test Missing Files process?"	

		<ul style="list-style-type: none"> - "Yes" - system end the process and redirect guest back to landing page. - "No" – Continue to find missing files.
Exceptions:		
No	Actor Action	System Response
1		System shows message the "Please check your connection" when the internet is lost.
2	Guest enter invalid URL.	System shows message "400 Bad request".

Relationship: N/A

Business Rules:

- Reasons why files become missing are shown in detail.
- With guest role, the limit of page test is set. System only tests five pages of the input website's address. For more pages, guest must register account and login to use full features.
- With guest role, the limit of file's format is set. System only check files that have .PNG and .JPEG extension. For more formats, guest must register account and login to use full features.

Table 9 - <Guest> Test missing file use case specification

2.3.2 <Staff> Overview Use Case

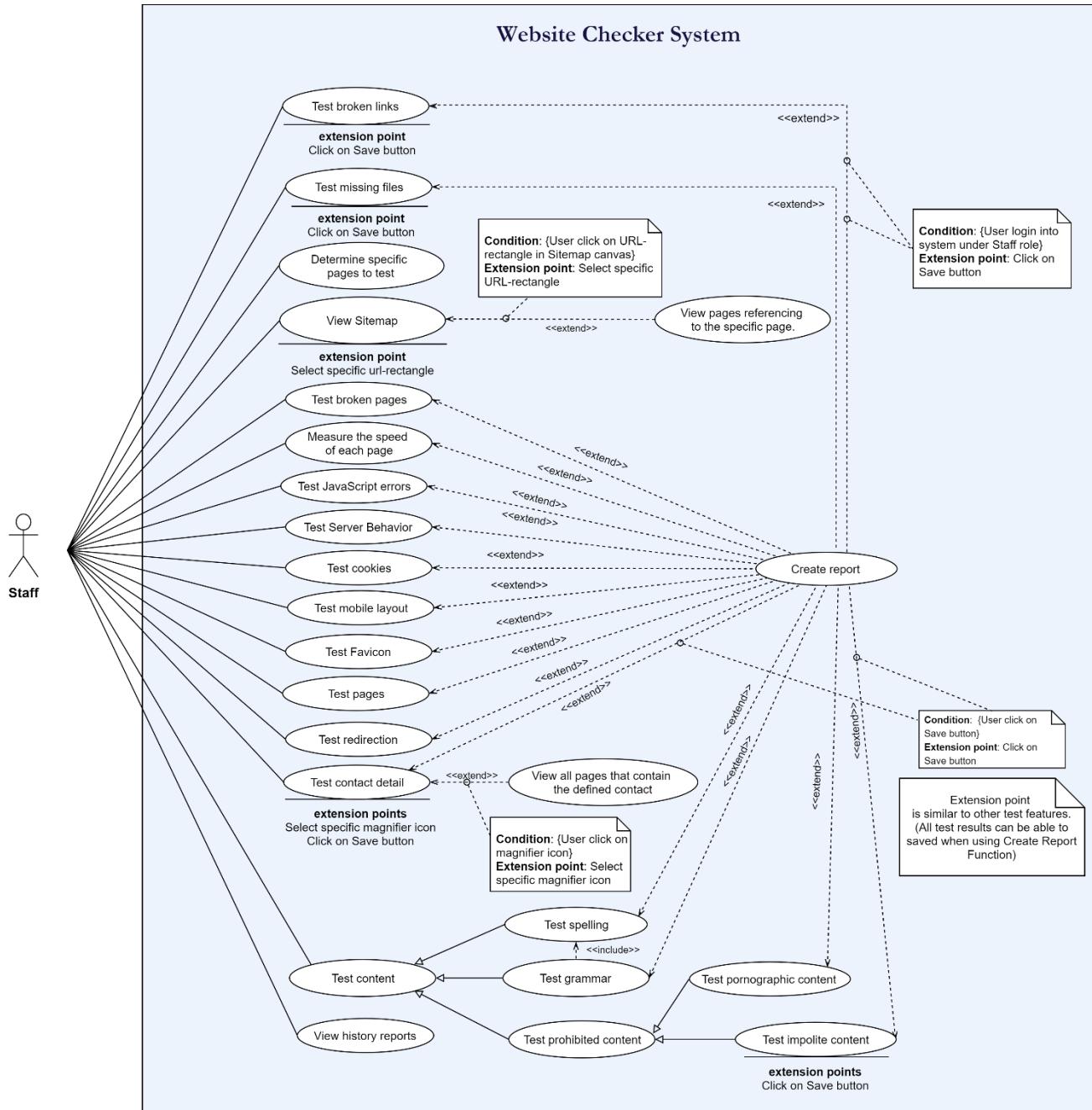


Figure 7 - Staff Overview Use Case

2.3.2.1 Determine specific pages to test



Figure 8-<Staff> Determine specific pages to test

USE CASE - WCS_UC_4			
Use Case No.	WCS_UC_4	Use Case Version	1.0
Use Case Name	Determine specific page to test		
Author	TruongLN		
Date	24/10/2018	Priority	High
Actor:			
- Staff			
Summary:			
- This use case allows staff to determine specific pages to test.			
Goal:			
- Staff can determine specific pages to test option.			
Triggers:			
- Staff sends determine specific pages command.			
Preconditions:			
- User must login into the system with role Staff.			
- Staff must be assigned for testing on this website by Manager.			
Post conditions:			
- Success: Display page option on screen.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff go to create page option screen.	Page option screen is shown with the following labels and fields: - Create Page Option : Plus button	
2	Staff click plus button.	Create page option screen is shown with the following labels and fields: - Page Option Name: text, required, length: 8-30.	
3	Staff inputs information.		
4	Staff send create page option command.	Determine specific page view is shown with the following fields: - Page url: toggle checkbox	
5	Staff chooses page to test.		
6	Staff send command to determine specific page.	Determine specific page to test in system.	

		[Exception 1]
Alternative Scenario:		
Exceptions:		
No	Actor Action	System Response
1	Determine specific page failed.	Show message to notify manager that remove failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.
Business rules:

- When website and sitemap must be created by manager
- At the first use, page option must default at root page
- The page which belong to website must be internal

Table 10 - <Staff> Determine specific pages to test use case specification

2.3.2.2 View sitemap

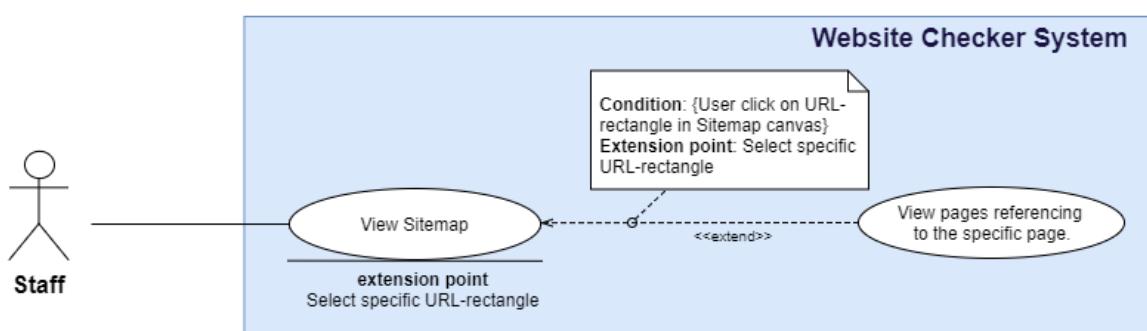


Figure 9 - <Staff> View sitemap

USE CASE – WCS_UC_5			
Use Case No.	WCS_UC_5	Use Case Version	1.0
Use Case Name	View Sitemap		
Author	ViNTT		
Date	24/10/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff to view sitemap. 		
Goal:	<ul style="list-style-type: none"> - Staff can get visual sitemap image on canvas, then they can view the detail references of each page. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends view sitemap command. 		

Preconditions:

- User must login into the system with role Staff.
- Staff must be assigned for testing at least one website by Manager.
- Staff must select the specific website to view sitemap.
- The sitemap structure must be built before by Manager.

Post conditions:

- Success: Display website sitemap canvas on screen.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to sitemap screen.	Show visual sitemap image on canvas screen. [Exceptions 1] [Exceptions 2]

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1		System shows error message "Please check your connection" when the internet is lost.
2		System show error message when the sitemap's data is not found for rendering.

Relationships: WCS_UC_1, WCS_UC_6, WCS_UC_24 to WCS_UC_26.

Business rules:

- The sitemap should be clearly visualized, supply user a general view about entire web structure.
- Staff can click on this visual image for more detail info. With each page, user easily know that page is referencing to which pages, or how many pages are referencing to it. By that way, user can ensure that there is no confusion in implement process.

Table 11 - <Staff> View Sitemap use case specification

2.3.2.3 View pages referencing to the specific page

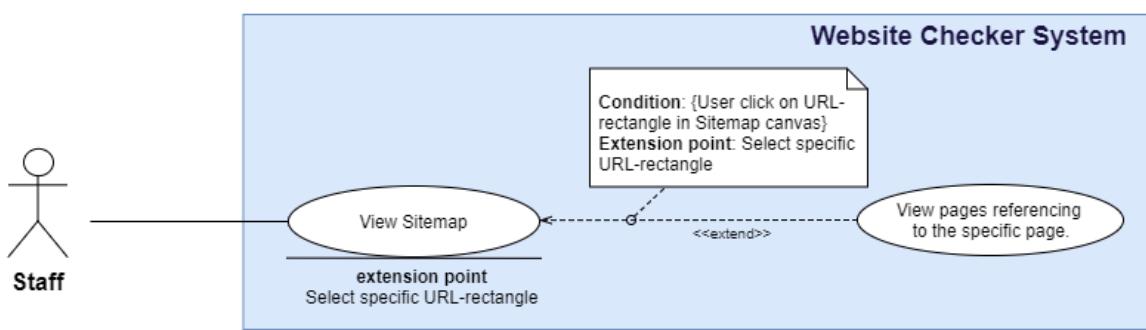


Figure 10 - <Staff> View pages referencing to the specific page

USE CASE - WCS_UC_6												
Use Case No.	WCS_UC_6	Use Case Version	1.0									
Use Case Name	View pages referencing to the specific page											
Author	ViNTT											
Date	24/10/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Staff 											
Summary:	<ul style="list-style-type: none"> - This use case allows staff to view pages referencing to the same page. 											
Goal:	<ul style="list-style-type: none"> - Staff can know redirect information of each page in details to check whether there is any confusion or not. 											
Triggers:	<ul style="list-style-type: none"> - Staff click on each URL rectangle in sitemap canvas. 											
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Staff. - Staff must be assigned for testing at least one website by Manager. - Staff must select the specific website to view sitemap. - The sitemap structure must be built before by Manager. 											
Post conditions:	<ul style="list-style-type: none"> - Success: Display the redirect information of selected URL in detail. - Fail: System shows error messages. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Staff go to sitemap screen.</td><td>Show visual sitemap image on canvas screen. [Exceptions 1] [Exceptions 2]</td></tr> <tr> <td>2</td><td>Staff click on URL rectangle in sitemap canvas.</td><td>Display the redirect information of selected URL in detail. [Exception 1, 2]</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Staff go to sitemap screen.	Show visual sitemap image on canvas screen. [Exceptions 1] [Exceptions 2]	2	Staff click on URL rectangle in sitemap canvas.	Display the redirect information of selected URL in detail. [Exception 1, 2]
Step	Actor Action	System Response										
1	Staff go to sitemap screen.	Show visual sitemap image on canvas screen. [Exceptions 1] [Exceptions 2]										
2	Staff click on URL rectangle in sitemap canvas.	Display the redirect information of selected URL in detail. [Exception 1, 2]										

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1		Show error message "Please check your connection" when the internet is lost.
2	If the URL not be valid anymore (or broken).	Show error message "System not received information of selected URL"

Relationships: WCS_UC_1, WCS_UC_5, WCS_UC_24 to WCS_UC_26.

Business Rules:

- With each URL selection of user, the system should let user chooses option: view selected page is referencing to which pages, or how many pages are referencing to it. Besides, the type of link is addition information: internal (backlink) or external.

Table 12 - <Staff> View pages referencing to the specific page use case specification

2.3.2.4 Test spelling for website's content

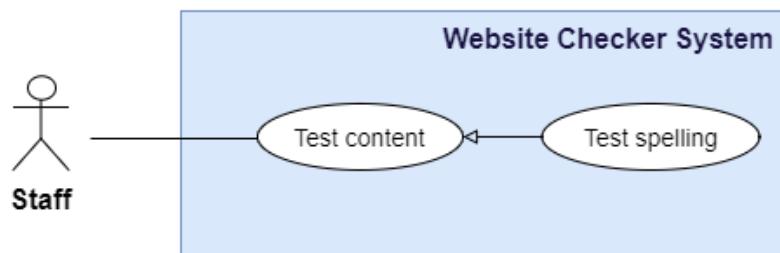


Figure 11 - <Staff> Test spelling

USE CASE - WCS_UC_7			
Use Case No.	WCS_UC_7	Use Case Version	1.0
Use Case Name	Test spelling		
Author	ViNTT		
Date	24/10/2018	Priority	High
Actor:			
- Staff			
Summary:			
- This use case allows staff to detect spelling mistake in web's content.			
Goal:			

- Staff can know which word is wrong then received recommend alternative words.

Triggers:

- Staff send test spelling command.

Preconditions:

- User must login into the system with role Staff.
- Staff must be assigned for testing at least one website by Manager.
- Staff must select the specific website and sub-pages to test.
- English dictionary must be predefined under DBMS.

Post conditions:

- Success: Display test result with alternative suggestions.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to Spelling screen.	Display screen including: <ul style="list-style-type: none"> - Table: contains previous test result. - Buttons: "Check" for sending test spelling command, "Save" for creating report and "Export" for exporting file in Excel format.
2	Staff sends test spelling for website's content command by clicking the "Check" button.	Display the result including: spelling mistakes with 3 recommend words. [Exception 1, 2]

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1		Show error message "Please check your connection" when the internet is lost.
2	Server cannot connect to DBMS.	Show error message "Dictionary is not available for test."

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_8, WCS_UC_22, WCS_UC_23, WCS_UC_37.

Business Rules:

- After giving the predicted words, system should give options for users: view suggestions, choose an alternative word, or confirm ignore those mistakes.

Ignore cases are saved. Next times, before giving suggestion, processor traverse from that list to decide to handle with detected mistakes.

- User's responding is collected and used for increasing the accuracy of spelling test processor. Chosen alternative words is marked with higher priority for giving suggestion results.

Table 13 - <Staff> Test spelling use case specification

2.3.2.5 Test grammar for website's content

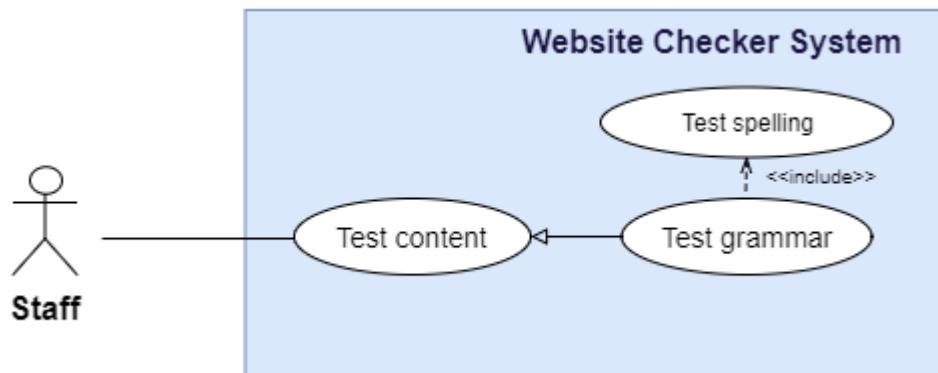


Figure 12 - <Staff> Test grammar

USE CASE – WCS_UC_8			
Use Case No.	WCS_UC_8	Use Case Version	1.0
Use Case Name	Test grammar		
Author	ViNTT		
Date	24/10/2018	Priority	High
Actor:			
- Staff			
Summary:			
- This use case allows staff to test grammar of website's content.			
Goal:			
- Staff can know where the grammar is wrong, reason why and how to correct those grammar mistakes.			
-			
Triggers:			
- Staff send test grammar command.			
Preconditions:			
- User must login into the system with role Staff.			
- Staff must be assigned for testing at least one website by Manager.			
- Staff must select the specific website to test grammar.			
Post conditions:			
- Success: Display the test result with grammar mistakes in detail.			
- Fail: System shows error messages.			

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to Grammar screen.	Display screen including: - Table: contains previous test result. Buttons: "Check" for sending test grammar command, "Save" for creating report and "Export" for exporting file in Excel format.
2	Staff send test grammar command by clicking on "Check" button.	Display the result including: fragments contain grammar mistakes, and reason make it wrong. [Exception 1, 2]

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1		Show error message "Please check your connection" when the internet is lost.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_7, WCS_UC_22, WCS_UC_23, WCS_UC_37.

Business Rules:

- Grammar test includes test spelling mistake to ensure there is no confusion.
- Reason why the grammar is wrong need to be displayed clearly with specific excerpt

Table 14 - <Staff> Test Grammar use case specification

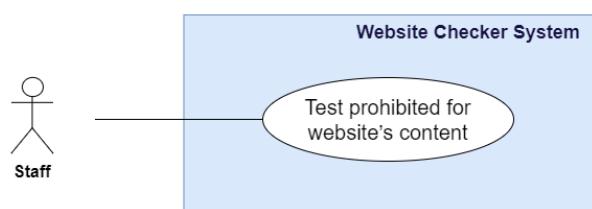
2.3.2.6 Test prohibited words for website's content

Figure 13 - <Staff> Test prohibited for website's content

USE CASE – WCS_UC_9

Use Case No.	WCS_UC_9	Use Case Version	1.0
Use Case Name	Test prohibited for website's content		
Author	TrinhND		
Date	24/10/2018	Priority	High

Actor:

- Staff

Summary:

- Allow Staff to test prohibited for website's content.

Goal:

- Staff can view prohibited content information: Word, Type, Fragment and Page.

Triggers:

- Staff sends test prohibited for website's content command.

Preconditions:

- User must login into the system with role Staff.
- Staff must be assigned for testing on this website by Manager.
- Staff must select pages which will be tested.
- The warning word list must be defined.

Post conditions:

- Success: Display the result test on screen, and Staff can create report also.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to Test prohibited content view.	System render test prohibited for website's content screen. [Alternative 1]
2	Staff sends test prohibited for website's content command.	System response the result with information: "Word", "Type", "Fragment", "Page". [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	System check the "Report History" in the data storage.	Render the result table that contains the previous test results with information: "Word", "Type", and "URL".

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection" when the internet is lost.

Relationships: WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28, WCS_UC_30.
Business Rules:

- Plural, past tense of warning word will also be checked.
- Warning word will be added by manager.
- All positions where the warning word appear are checked
- Fragment include prohibited will be show.
- Staff can create report about result test.
- Staff can export result test to excel file.

Table 15 - <Staff> Test prohibited for website's content use case specification

2.3.2.7 Test broken links

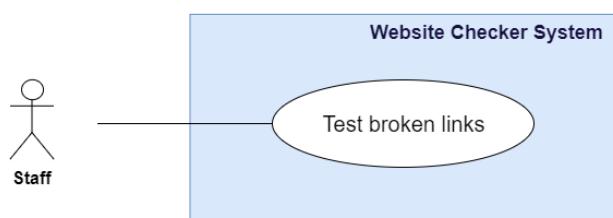


Figure 14 - <Staff> Test broken links

USE CASE – WCS_UC_10			
Use Case No.	WCS_UC_10	Use Case Version	1.0
Use Case Name	Test broken links		
Author	TrinhND		
Date	24/10/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - Allow Staff to test broken links of a website. 		
Goal:	<ul style="list-style-type: none"> - Staff can view broken links information: page URL, broken link URL, and status code. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends test broken links command. 		

Preconditions:

- User must login into the system with role Staff.
- Staff must be assigned for testing on this website by Manager.
- Staff must select pages which will be tested.

Post conditions:

- Success: Display the result test on screen, and Staff can create report also.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to Test broken links view.	System render test broken links screen. [Alternative 1]
2	Staff sends test broken links command.	System response the result with information: "Page", "Broken Link", "Issue". [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	System check the "Report History" in the data storage.	Render the result table that contains the previous test results with information: "Page", "Broken Link", and "Issue".

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection" when the internet is lost.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Test both internal links and external links of the website.
- Only HTTP_NOT_FOUND and HTTP_BAD_REQUEST will be checked.
- Issue that make broken links will be shown in detail.
- Staff can create report about result test.
- Staff can export result test to excel file.

Table 16 - <Staff> Test broken links use case specification

2.3.2.8 Test broken pages

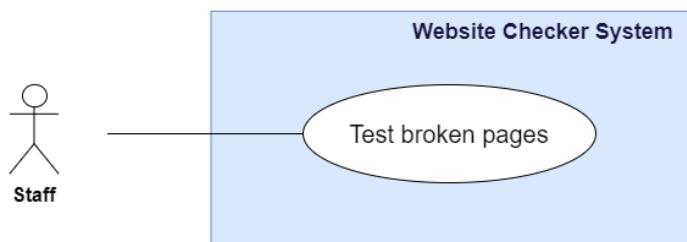


Figure 15 - <Staff> Test broken pages

USE CASE – WCS_UC_11			
Use Case No.	WCS_UC_11	Use Case Version	1.0
Use Case Name	Test broken pages		
Author	TrinhND		
Date	24/10/2018	Priority	High
Actor:			
- Staff			
Summary:			
- Allow Staff to test broken pages of a website.			
Goal:			
- Staff can view broken pages information: page URL, status code and response message.			
Triggers:			
- Staff sends test broken pages command.			
Preconditions:			
- User must login into the system with role Staff.			
- Staff must be assigned for testing on this website by Manager.			
- Staff must select pages which will be tested.			
Post conditions:			
- Success: Display the result test on screen, and Staff can create report also.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff goes to Test broken pages view.	System render test broken pages screen. [Alternative 1]	
2	Staff sends test broken pages command.	System response the result with information: "Page", "Status", "Issue". [Exception 1]	
Alternative Scenario:			

No	Actor Action	System Response
1	System check the "Report History" in the data storage.	Render the result table that contains the previous test results with information: "Page", "Status", and "Issue".

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection" when the internet is lost.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Test only internal pages of the website.
- Status code of Success (2xx), Client errors (4xx), Server errors (5xx) will be checked.
- Issue that make broken pages will be shown in detail.
- Staff can create report about result test.
- Staff can export result test to excel file.

Table 17 - <Staff> Test broken pages use case specification

2.3.2.9 Test missing files

Figure 16 - <Staff> Test missing file

USE CASE - WCS_UC_12			
Use Case No.	WCS_UC_12	Use Case Version	1.0
Use Case Name	Test Missing File		
Author	AnhTP		
Date	24/10/2018	Priority	High

Actor:

- Staff

Summary:

- This use case allows staff to detect missing file with the defined website's address.

Goal:

- Staff can view missing file information: file URL source, description missing of each page on the result screen.

Triggers:

- Staff sends test missing file command.

Preconditions:

- User must login into the system with role Staff.
- Staff must be assigned for testing on this website by Manager.
- Staff must select pages which will be tested.

Post conditions:

- Success: Display the result test on screen.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to missing file screen.	System render the missing file screen.
2	Staff send command to test missing file.	System response the result with information: "File", "Description", "Page affected". [Exception 1, 2]

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Tested missing file fail	Show message to notify manager that test missing file failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Reasons that files become missing are shown in detail.
- Missing file is a file cannot connect or file size is 0.

- Staff can test all type file or choose type file for testing missing file: image, CSS, doc, archives.
- System only check files image that have .png, .jpeg, .jpg extension.
- System only check files CSS that have .css extension.
- System only check files DOC that have .doc, .pdf, .ppt, xls extension.
- System only check files archives that have .rar, .jar, .zip extension.
- System only check files video that have .mp4 extension.
- System only check files audio that have .mp3 extension.
- After the missing file test is finished, staff can save and export to excel the result.

Table 18 - <Staff> Test missing file use case specification

2.3.2.10 Test the mobile layout

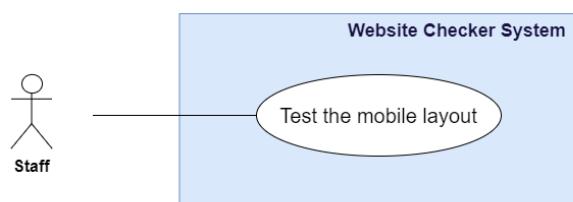


Figure 17 - <Staff> Test the mobile layout

USE CASE - WCS_UC_13			
Use Case No.	WCS_UC_13	Use Case Version	1.0
Use Case Name	Test the mobile layout		
Author	TrinhND		
Date	24/10/2018	Priority	High
Actor:			
- Staff			
Summary:			
- Allow Staff to test mobile layout of each page in website.			
Goal:			
- Staff can view mobile layout information: Screenshot, URL page, and Issue.			
Triggers:			
- Staff sends test the mobile layout command.			
Preconditions:			
<ul style="list-style-type: none"> - User must login into the system with role Staff. - Staff must be assigned for testing on this website by Manager. - Staff must select pages which will be tested. 			
Post conditions:			
<ul style="list-style-type: none"> - Success: Display the result test on screen, and Staff can create report also. - Fail: System shows error messages. 			

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to Test broken pages view.	System render test broken pages screen. [Alternative 1]
2	Staff sends test broken pages command.	System response the result with information: "Page", "Status", "Issue". [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	System check the "Report History" in the data storage.	Render the result table that contains the previous test results with information: "Screenshot", "Page", and "Issue".

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection" when the internet is lost.
2		System show alt of Screenshot when the Screenshot can't load.

Relationships: WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Standard of the mobile layout will be defined base on W3 recommendation.
- Best font-size is 16px, touch-size is 44x44.
- Check viewport meta tag, manifest, mobile version, framework.
- Check support pinch to zoom, scroll horizontally.
- The test will be simulated in a mobile device and take a screenshot after load.
- Staff can create report about result test.
- Staff can export result test to excel file.

Table 19-<Staff> Test the mobile layout use case specification

2.3.2.11 Measure the speed test

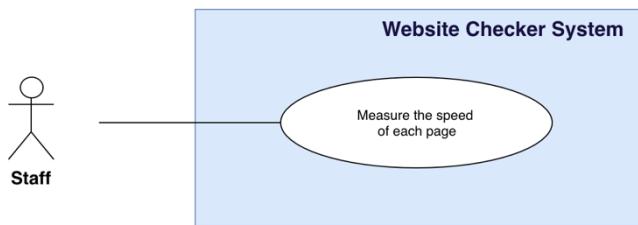


Figure 18 - <Staff> Measure the speed test of each page

USE CASE - WCS_UC_14			
Use Case No.	WCS_UC_14	Use Case Version	1.0
Use Case Name	Measure the speed of each page		
Author	TruongLN		
Date	24/10/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff to measure the speed of each page with a defined page's address. 		
Goal:	<ul style="list-style-type: none"> - Staff can view measure the speed of each page information: Interactive time, page load time and page size on the result screen. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends measure the speed of each page command. 		
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Staff. - Staff must be assigned for testing on this website by Manager. - Staff must select pages which will be tested. 		
Post conditions:	<ul style="list-style-type: none"> - Success: Display the result test on screen. - Fail: System shows error messages. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff go to measure the speed of each page screen.	System render the measure the speed of each page screen.	
2	Staff send command to test measure the speed of each page.	System response the result with information: "URL", "Interactive time", "page load time", "page size". [Exception 1]	

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Measured the speed of each page fail.	Show message to notify manager that measure the speed of each page failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- The page interactive time must be lower than the page load time.
- The result page size must be in first time load without cache.

Table 20 - <Staff> Measure the speed of each page use case specification

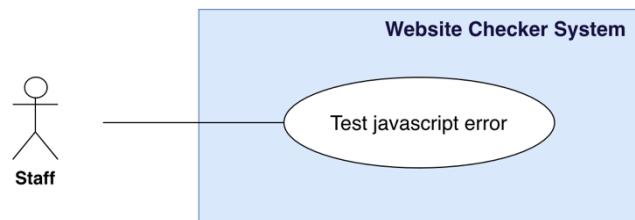
2.3.2.12 Test JavaScript errors

Figure 19 - <Staff> Test JavaScript errors

USE CASE – WCS_UC_15

Use Case No.	WCS_UC_15	Use Case Version	1.0
Use Case Name	Test JavaScript errors		
Author	TruongLN		
Date	24/10/2018	Priority	High

Actor:

- Staff

Summary:

- This use case allows staff to detect JavaScript errors with a defined page's address.

Goal:

- Staff can view JavaScript errors information: error message, type on the result screen.

Triggers:

- Staff sends test JavaScript errors command.

Preconditions:

- User must login into the system with role Staff.

- Staff must be assigned for testing on this website by Manager.
- Staff must select pages which will be tested.

Post conditions:

- Success: Display the result test on screen.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to javascript errors screen.	System render the test javascript errors screen.
2	Staff send command to test javascript errors.	System response the result with information: "Error Message", "Type", "Page". [Exception 1]

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Test javascript error fail.	Show message to notify manager that test javascript error failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Test the website page has javascript errors or not
- Every error must show error line and error javascript file.

Table 21 - <Staff> Test JavaScript errors use case specification

2.3.2.13 Test Favicon



Figure 20-<Staff> Test favicon

USE CASE – WCS_UC_16

Use Case No.	WCS_UC_16	Use Case Version	1.0
--------------	-----------	------------------	-----

Use Case Name	Test Favicon		
Author	AnhTP		
Date	24/10/2018	Priority	High

Actor:

- Staff

Summary:

- This use case allows staff to detect favicon with the defined website's address.

Goal:

- Staff can view favicon information: favicon image, favicon URL source, favicon size and type favicon on the result screen.

Triggers:

- Staff sends test favicon command.

Preconditions:

- User must login into the system with role Staff.
- Staff must be assigned for testing on this website by Manager.
- Staff must select pages which is tested.

Post conditions:

- Success: Display the result test on screen.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to favicon screen.	System render the favicon screen.
2	Staff send command to test favicon.	System response the result with information: "Favicon", "URL", "Width", "Type", "Page". [Exception 1].

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Test Favicon fail.	Show message to notify manager that test favicon failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Test the website page has favicon or not.
- System only check favicon two ways:
 - o Check a rel attribute value defined in a profile.

- Check the favicon at a predefined URI.
- Number of favicon different and number pages missing favicon
- If the website page has the favicon, the result display the favicon information.
- If the website pages has not the favicon, the result display missing favicon message.
- After the favicon test is finished, staff can save and export to excel the result.

Table 22- <Staff> Test favicon use case specification

2.3.2.14 Test cookie website using

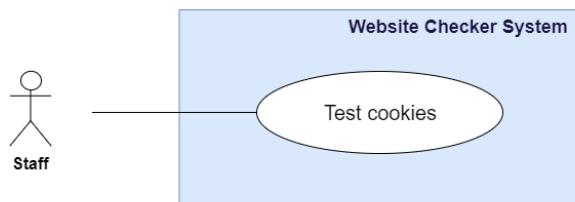


Figure 21 - <Staff> Test cookies

USE CASE – WCS_UC_17			
Use Case No.	WCS_UC_17	Use Case Version	1.0
Use Case Name	Test cookies		
Author	TrinhND		
Date	24/10/2018	Priority	High
Actor:			
- Staff			
Summary:			
- Allow Staff to test cookies be used in website.			
Goal:			
- Staff can view cookies information: Name, Category, Party and Description.			
Triggers:			
- Staff sends test cookies command.			
Preconditions:			
- User must login into the system with role Staff.			
- Staff must be assigned for testing on this website by Manager.			
- Staff must select pages which will be tested.			
Post conditions:			
- Success: Display the result test on screen, and Staff can create report also.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff goes to Test broken links view.	System render test cookies screen.	

		[Alternative 1]
2	Staff sends test cookies command.	System response the result with information: "Name", "Category", "Party", "Description". [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	System check the "Report History" in the data storage.	Render the result table that contains the previous test results with information: "Name", "Category", "Party", and "Description".

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection" when the internet is lost.

Relationships: WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Cookies will be checked base on EU Cookie Law, description and party of popular cookies will be show in detail.
- Popular cookies are provided by Google, Microsoft, Addthis.com, doubleclick.net.
- All cookies used will be shown.
- Staff can create report about result test.
- Staff can export result test to excel file.

Table 23 - <Staff> Test cookies use case specification

2.3.2.15 Test server behavior

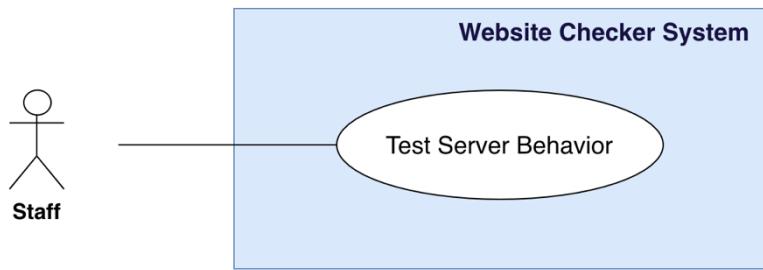


Figure 22 - <Staff> Test server behavior

USE CASE - WCS_UC_18												
Use Case No.	WCS_UC_18	Use Case Version	1.0									
Use Case Name	Test Server Behavior											
Author	TruongLN											
Date	24/10/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Staff 											
Summary:	<ul style="list-style-type: none"> - This use case allows staff to detect server behavior. 											
Goal:	<ul style="list-style-type: none"> - Staff can view server behavior information: website redirects from www to non-www and HTTP to HTTPS, HTTPS, website has a working page “Page Not Found” page, all pages use SSL on the result screen. 											
Triggers:	<ul style="list-style-type: none"> - Staff sends test server behav command. 											
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Staff. - Staff must be assigned for testing on this website by Manager. - Staff must select pages which will be tested. 											
Post conditions:	<ul style="list-style-type: none"> - Success: Display the result test on screen. - Fail: System shows error messages. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Staff go to server behavior screen.</td><td>System render the server behavior screen.</td></tr> <tr> <td>2</td><td>Staff send command to test favicon.</td><td>System response the result with information: “website redirects from www to non-www”,</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Staff go to server behavior screen.	System render the server behavior screen.	2	Staff send command to test favicon.	System response the result with information: “website redirects from www to non-www”,
Step	Actor Action	System Response										
1	Staff go to server behavior screen.	System render the server behavior screen.										
2	Staff send command to test favicon.	System response the result with information: “website redirects from www to non-www”,										

		“ Website redirects from HTTP to HTTPS”, “ All pages use SSL ”. [Exception 1, 2]
--	--	---

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Test server behavior fail	Show message to notify manager that test server behavior failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.**Business Rules:**

- Test the website page has redirects from www to non www or from HTTP to HTTPS.
- Test All pages have SSL certification.

Table 24 - <Staff> Test server behavior use case specification

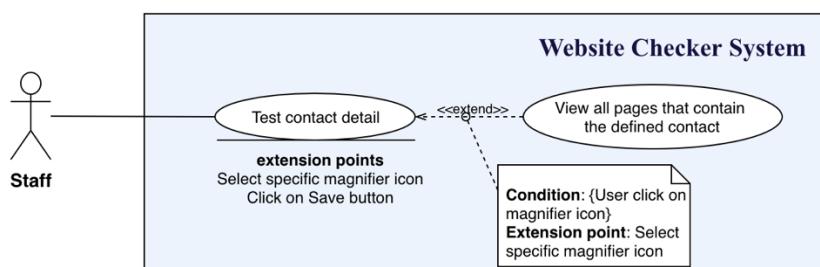
2.3.2.16 Test contact details on the website

Figure 23 - <Staff> Test contact details on the website

USE CASE – WCS_UC_19			
Use Case No.	WCS_UC_19	Use Case Version	1.0
Use Case Name	Test Contact Details On The Website		
Author	AnhTP		
Date	24/10/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff to detect contact detail on page with the defined website's address. 		
Goal:	<ul style="list-style-type: none"> - Staff can view contact detail information: Contact, Page Affected on the result screen. 		
Triggers:			

- Staff sends test redirection command.

Preconditions:

- User must login into the system with role Staff.
- Staff must be assigned for testing on this website by Manager.
- Staff must select pages which is tested.

Post conditions:

- Success: Display the result test on screen.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to test contact details screen.	System render the contact detail screen.
2	Staff send command to test contact detail.	System response the result with information: "Contact", "Page Affected", "Action". [Exception 1]

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Test contact detail fail.	Show message to notify manager that test contact detail failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Test the website page has contact detail or not.
- Staff can check the contact detail is correctly on the website page.
- Test contact details with pattern:
 - o Phone: pattern international phone number (example: +84 89 8000 8900).
 - o Email: pattern email (example: example@gmail.com).
- If the website page has contact detail, the result display the contact details information.
- After the contact detail test is finished, staff can save and export to excel the result.
- Staff can click magnifier and view all pages that contain the contact defined.

Table 25 - <Staff> Test contact details on the website use case specification

2.3.2.17 View all pages that contain the contact defined

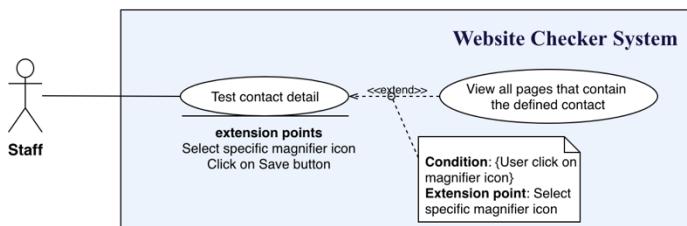


Figure 24 - <Staff> View all pages that contain the contact defined

USE CASE - WCS_UC_20			
Use Case No.	WCS_UC_20	Use Case Version	1.0
Use Case Name	View all pages that contain that contact defined		
Author	AnhTP		
Date	24/10/2018	Priority	Normal
Actor:			
- Staff			
Summary:			
- This use case allows staff to view all pages that contains that contact detail defined.			
Goal:			
- Staff can view all pages that contains the contact detail defined information: list of pages on the result screen.			
Triggers:			
- Staff click on magnifier icon.			
Preconditions:			
- User must login into the system with role Staff.			
- Staff must be assigned for testing on this website by Manager.			
Post conditions:			
- Success: Display the result test on screen.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff go to test contact details screen.	System render the contact detail screen.	
2	Staff click on magnifier icon with the contact detail defined.	System response the result with information: "Page Affected". [Exception 1]	

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	View all pages that contains the contact detail defined fail	Show message to notify manager that view all pages that contains the contact detail defined failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Test contact detail is tested.
- Show list of pages that contains the contact detail defined.

Table 26 - <Staff> View all pages that contain the contact defined use case specification

2.3.2.18 Test redirections

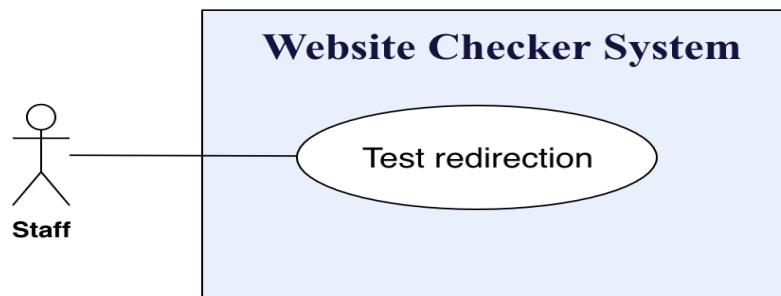


Figure 25 - <Staff> Test redirection

USE CASE – WCS_UC_21

Use Case No.	WCS_UC_21	Use Case Version	1.0
Use Case Name	Test Redirection		
Author	AnhTP		
Date	24/10/2018	Priority	High

Actor:

- Staff

Summary:

- This use case allows staff to detect page redirection with the defined website's address.

Goal:

- Staff can view page redirection information: web address, directs to URL, type redirection, http code response on the result screen.

Triggers:

- Staff sends test redirection command.

Preconditions:

- User must login into the system with role Staff.
- Staff must be assigned for testing on this website by Manager.
- Staff must select pages which will be tested.

Post conditions:

- Success: Display the result test on screen.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to redirection screen.	System render the redirection screen.
2	Staff send command to test redirection.	System response the result with information: "web address", "directs to url", "type", "HTTP Code". [[Exception 1]]

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Tested redirections fail.	Show message to notify manager that test redirection failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Test the website page is redirection.
- Type redirection is traditional reidrection type: HTTP 301 Moved Permanently and HTTP 302 Found
- If the website page is redirection, the result display the redirection information.
- If the website pages is not redirection, the result display not redirection found.
- After the redirection test is finished, staff can save and export to excel the result.

Table 27 - <Staff> Test redirection use case specification

2.3.2.19 Test website's pages

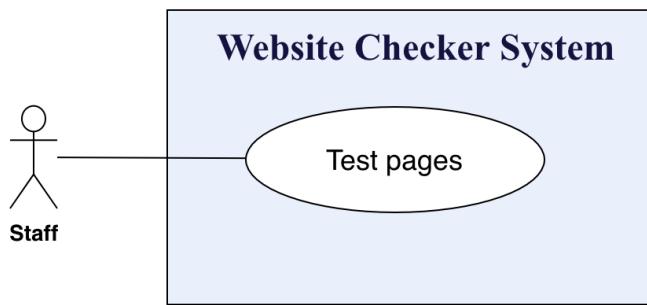


Figure 26 - <Staff> Test pages

USE CASE – WCS_UC_22			
Use Case No.	WCS_UC_22	Use Case Version	1.0
Use Case Name	Test Pages		
Author	AnhTP		
Date	24/10/2018	Priority	High
Actor:			
- Staff			
Summary:			
- This use case allows staff to detect website's page with the defined website's address.			
Goal:			
- Staff can view page information: page title, web address, canonical url, http code response on the result screen.			
Triggers:			
- Staff sends test pages command.			
Preconditions:			
- User must login into the system with role Staff.			
- Staff must be assigned for testing on this website by Manager.			
- Staff must select pages which will be tested.			
Post conditions:			
- Success: Display the result test on screen.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff go to page test screen.	System render the page test screen.	
7	Staff send command to test pages.	System response the result with information: "Title", "web address", "Canonical url", "HTTP code". [Exception 1]	

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Tested pages fail.	Show message to notify manager that test page failed.

Relationships: WCS_UC_1, WCS_UC_4, WCS_UC_22, WCS_UC_23, WCS_UC_28.

Business Rules:

- Test the website page status.
- The website page has title or not.
- The website page has canonical URL or not.
- The result display the page information.
- After the page test is finished, staff can save and export to excel the result.

Table 28 - <Staff> Test pages use case specification

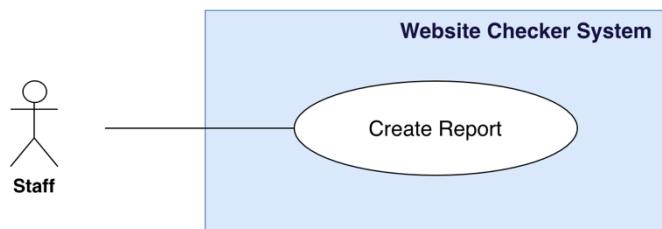
2.3.2.20 Create report

Figure 27 <Staff> Create report

USE CASE – WCS_UC_23

Use Case No.	WCS_UC_23	Use Case Version	1.0
Use Case Name	Create Report		
Author	TrinhND		
Date	24/10/2018	Priority	Normal

Actor:

- Staff

Summary:

- Allow Staff to create report after do each test function.

Goal:

- Staff can create report after Staff do each test function and save in “Report History”.

Triggers:

- Staff sends create report command.

Preconditions:

- User must login into the system with role Staff.
- Test function must be done.

Post conditions:

- Success: New report about tested function will be create and save.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to “result test” view.	System render result test screen.
2	Staff sends Create Report command.	Implement Create Report feature. [Exception 1]
3		Save the report to the “Report History”.

Alternative Scenario: N/A

Exceptions:

No	Actor Action	System Response
1	No test function completed yet.	Show message “The test function must be complete”.

Relationships: WCS_UC_4, WCS_UC_5, WCS_UC_6... to WCS_UC_31.

Business Rules:

- New report will be create and save into “Report History”.
- Report can be save after each function test.
- At least one function test has been done and show the result.
- A notification will be sent to Staff after the process is completed.

Table 29 - <Staff> Create report use case specification

2.3.2.21 View history reports

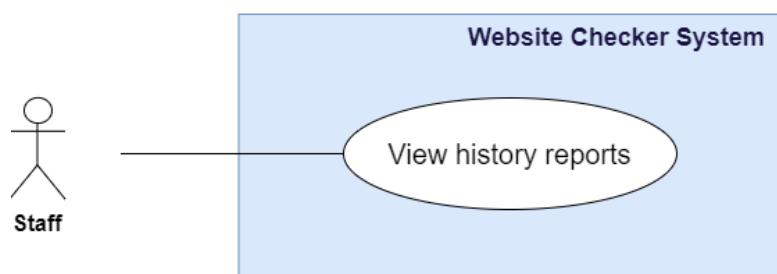


Figure 28 - <Staff> View history reports

USE CASE - WCS_UC_24

Use Case No.	WCS_UC_24	Use Case Version	1.0
Use Case Name	View History Reports		
Author	TrinhND		
Date	24/10/2018	Priority	Normal

Actor:

- Staff

Summary:

- Allow Staff to view all history reports have been created.

Goal:

- Staff can view all reports have been created.

Triggers:

- Staff sends view history reports command.

Preconditions:

- User must login into the system with role Staff.

Post conditions:

- Success: History report will be show by date created.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff sends view history reports command.	System render view history reports screen. [Alternative 1]
2	Staff send date time picker command.	Show history report table by day was chosen with information: "No", "Test Function", "Time" [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1		Show history report table by latest date with information: "No", "Test Function", "Time"

Exceptions:

No	Actor Action	System Response
1	Staff choose a day without report.	History report table is empty.

Relationships: WCS_UC_1, WCS_UC_7... to WCS_UC_22, WCS_UC_28.

Business Rules:

- At least one report should be done and save.
- History report will be display and filter by time created.
- Staff can choose day to view history report.

Table 30 <Staff> View history reports use case specification

2.3.3 <Manager> Overview Use Case

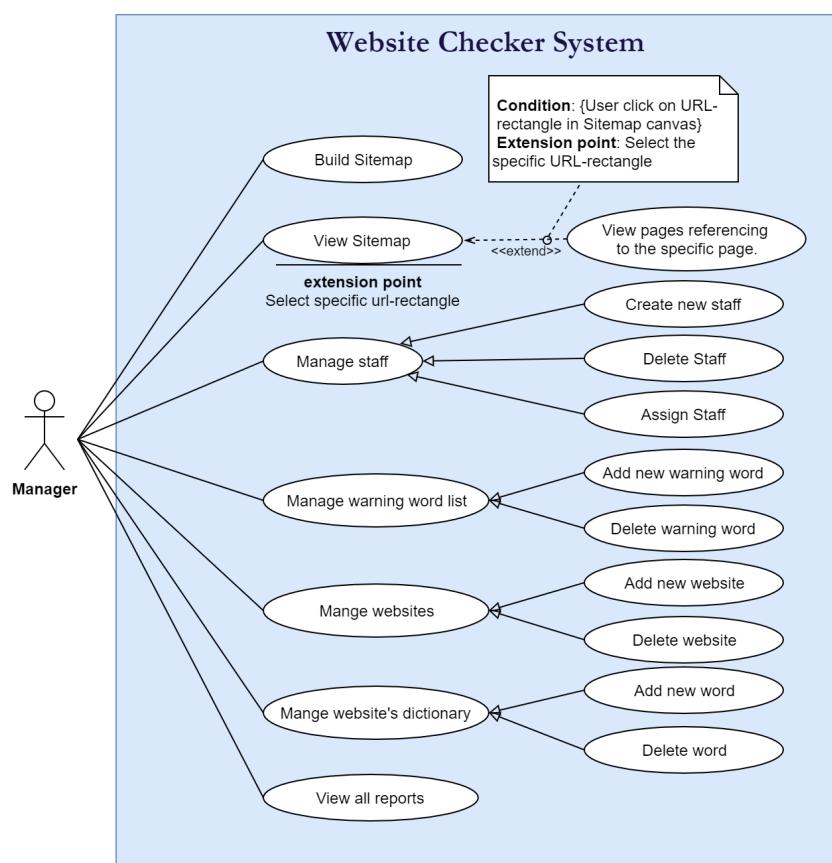


Figure 29 - Manager Overview Use Case

2.3.3.1 Build Sitemap

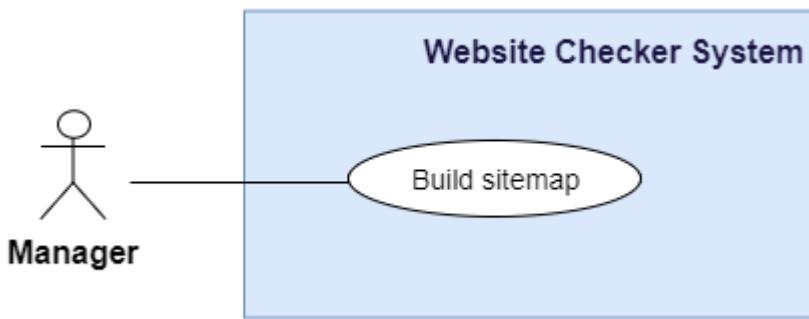


Figure 30 - <Manager> Build Sitemap

USE CASE - WCS_UC_24									
Use Case No.	WCS_UC_24	Use Case Version	1.0						
Use Case Name	Build Sitemap								
Author	ViNTT								
Date	24/10/2018	Priority	High						
Actor:	<ul style="list-style-type: none"> - Manager 								
Summary:	<ul style="list-style-type: none"> - This use case allows manager to build sitemap. 								
Goal:	<ul style="list-style-type: none"> - Manager can build or update the link structure of website with latest version. 								
Triggers:	<ul style="list-style-type: none"> - Manager sends build sitemap command. 								
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Manager. - Staff must select the specific website to build sitemap. 								
Post conditions:	<ul style="list-style-type: none"> - Success: Display success message. - Fail: System shows error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Manager goes to Manage Website screen.</td><td> <p>Show table with website information list, addition with 2 buttons related sitemap:</p> <ul style="list-style-type: none"> - “Build”: button using for build/update version. - “View sitemap”: button using for sitemap information in detail. </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Manager goes to Manage Website screen.	<p>Show table with website information list, addition with 2 buttons related sitemap:</p> <ul style="list-style-type: none"> - “Build”: button using for build/update version. - “View sitemap”: button using for sitemap information in detail.
Step	Actor Action	System Response							
1	Manager goes to Manage Website screen.	<p>Show table with website information list, addition with 2 buttons related sitemap:</p> <ul style="list-style-type: none"> - “Build”: button using for build/update version. - “View sitemap”: button using for sitemap information in detail. 							

2	Manager sends build sitemap command.	Show "Successfully completed" message. [Exception 1, 2]
---	--------------------------------------	---

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1		System shows error message "Please check your connection" when the internet is lost.
2		System shows error message when the system cannot get the link structure.

Relationships: WCS_UC_1, WCS_UC_6, WCS_UC_25, WCS_UC_26.**Business Rules:**

- The link structure of website is collected and organized as directed graph.
- Graph is implemented as a list of list of <SiteLinkInfo> where SiteLinkInfo contains data: sourceURL, destinationURL, and the type of that destination URL.
- A graph keeps reference information in the direction from it (current URL) to another URL.
- An inverse graph keeps inverse direction: others URL direct to the same URL.
- Using Breadth-First Search combined queue, mark appeared vertices before to avoid infinite loop when display relationship between sub-pages.
- A HashSet with key-value pair contains: key is URL, value is an integer number (also is index number of this vertice in graph).
- The built sitemap will be decoded to use for render in UI. The sitemap's structure is stored in json format and save to database for next accessing times.
- The type of links is categorized: internal, external. Only concerned about internal links of website, external's references do not need to stored.

Table 31 - <Manager> Build sitemap use case specification

2.3.3.2 View sitemap

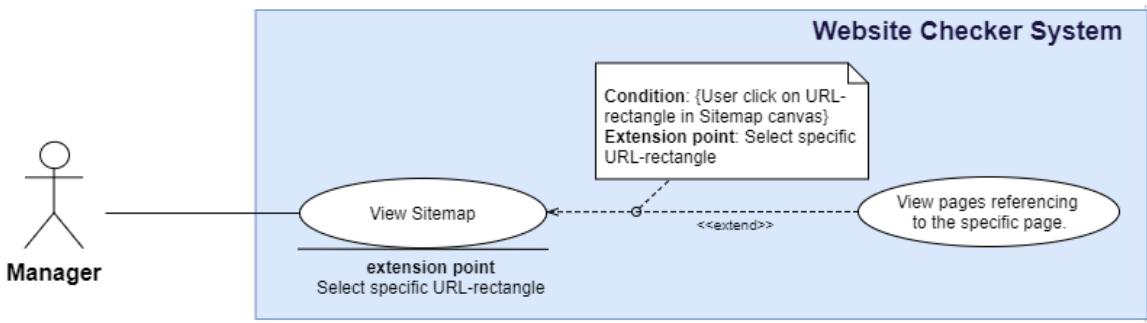


Figure 31 - <Manager> View sitemap

USE CASE – WCS_UC_25									
Use Case No.	WCS_UC_25	Use Case Version	1.0						
Use Case Name	View Sitemap								
Author	ViNTT								
Date	24/10/2018	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - Manager 								
Summary:	<ul style="list-style-type: none"> - This use case allows manager to view sitemap. 								
Goal:	<ul style="list-style-type: none"> - Manager can get visual sitemap image on canvas, then they can view the detail references of each page. 								
Triggers:	<ul style="list-style-type: none"> - Manager sends view sitemap command. 								
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Manager. - Staff must select the specific website to view sitemap. - The sitemap structure must be built before by Manager. 								
Post conditions:	<ul style="list-style-type: none"> - Success: Display website sitemap canvas on screen. - Fail: System shows error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Manager go to Manage Website screen.</td><td>Show visual sitemap image on canvas screen. [Exceptions 1] [Exceptions 2]</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Manager go to Manage Website screen.	Show visual sitemap image on canvas screen. [Exceptions 1] [Exceptions 2]
Step	Actor Action	System Response							
1	Manager go to Manage Website screen.	Show visual sitemap image on canvas screen. [Exceptions 1] [Exceptions 2]							
Alternative Scenario:									
Exceptions:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> </table>			No	Actor Action	System Response			
No	Actor Action	System Response							

1		System shows error message "Please check your connection" when the internet is lost.
2		System show error message when the sitemap's data is not found for rendering.

Relationships: WCS_UC_1, WCS_UC_6, WCS_UC_24 to WCS_UC_26.

Business Rules:

- The sitemap should be clearly visualized, supply user a general view about entire web structure.
- Manager can click on this visual image for more detail info.
- With each page, user easily know that page is referencing to which pages, or how many pages are referencing to it. By that way, user can ensure that there is no confusion in implement process.

Table 32 - <Manager> View sitemap use case specification

2.3.3.3 View pages referencing to the specific page

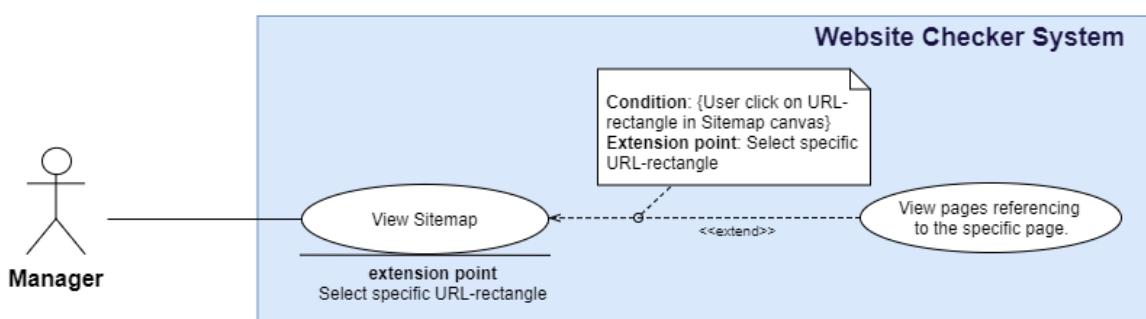


Figure 32 - <Manager> View pages referencing on the specific page

USE CASE - WCS_UC_26			
Use Case No.	WCS_UC_26	Use Case Version	1.0
Use Case Name	View pages referencing to the specific page		
Author	ViNTT		
Date	24/10/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Manager 		
Summary:	<ul style="list-style-type: none"> - This use case allows manager to view pages referencing to the same page. 		
Goal:			

- Manager can know redirect information of each page in details to check whether there is any confusion or not.

Triggers:

- Manager click on each URL rectangle in sitemap canvas.

Preconditions:

- User must login into the system with role Manager.
- Staff must select the specific website to view sitemap.
- The sitemap structure must be built before by Manager.

Post conditions:

- Success: Display the redirect information of selected URL in detail.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Manager goes to sitemap screen.	Show visual sitemap image on canvas screen. [Exceptions 1] [Exceptions 2]
2	Manager clicks on URL rectangle in sitemap canvas.	Display the redirect information of selected URL in detail. [Exception 1, 2]

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1		Show error message "Please check your connection" when the internet is lost.
2	If the URL not be valid anymore (or broken).	Show error message "System not received information of selected URL"

Relationships: WCS_UC_1, WCS_UC_5, WCS_UC_24 to WCS_UC_25.

Business Rules:

- With each URL selection of user, the system should let user chooses option: view selected page is referencing to which pages, or how many pages are referencing to it. Besides, the type of link is addition information: internal (backlink) or external.

Table 33 - <Manager> View pages referencing on the specific page use case specification

2.3.3.4 Add new website

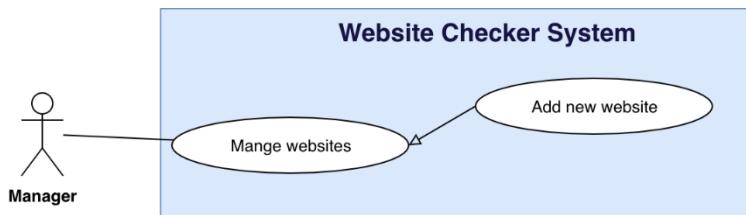


Figure 33 - <Manager> Add new website

USE CASE - WCS_UC_28												
Use Case No.	WCS_UC_28	Use Case Version	1.0									
Use Case Name	Add new website											
Author	AnhTP											
Date	24/10/2018	Priority	Normal									
Actor:	<ul style="list-style-type: none"> - Manager 											
Summary:	<ul style="list-style-type: none"> - This use case allows manager to add a new website with the defined website's address. 											
Goal:	<ul style="list-style-type: none"> - Manager successfully adds new website. 											
Triggers:	<ul style="list-style-type: none"> - Manager send add new website command to the system. 											
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Manager. 											
Post conditions:	<ul style="list-style-type: none"> - Success: New website successful store in database. - Fail: New website cannot be saved. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Manager goes to add new website screen.</td><td> Add new website screen is shown with following labels and fields: <ul style="list-style-type: none"> • Name website: text, required, length 8-30. • Website URL: text, required, length 8-30. </td></tr> <tr> <td>2</td><td>Manager inputs information.</td><td></td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Manager goes to add new website screen.	Add new website screen is shown with following labels and fields: <ul style="list-style-type: none"> • Name website: text, required, length 8-30. • Website URL: text, required, length 8-30. 	2	Manager inputs information.	
Step	Actor Action	System Response										
1	Manager goes to add new website screen.	Add new website screen is shown with following labels and fields: <ul style="list-style-type: none"> • Name website: text, required, length 8-30. • Website URL: text, required, length 8-30. 										
2	Manager inputs information.											

3	Manager send add new website command.	Add new website in the system. [Exception 1] System shows message website has been added successfully.
---	---------------------------------------	--

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Add new website fail	Show message to notify manager that add new website failed.

Relationships: WCS_UC_1.**Business Rules:**

- In case of success scenarios, a new website is added in the system.
- Website URL has pattern http:// or https://.
- Website URL is not duplicate.

Table 34 - <Manager> Add new website use case specification

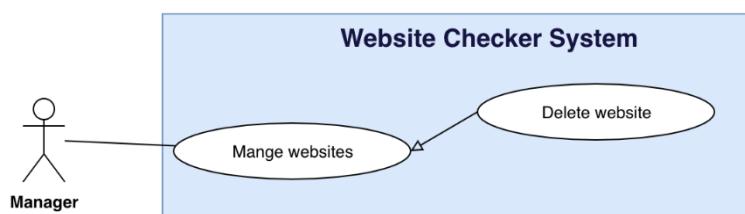
2.3.3.5 Delete website

Figure 34 <Manager> Delete website

USE CASE – WCS_UC_29			
Use Case No.	WCS_UC_29	Use Case Version	1.0
Use Case Name	Delete website		
Author	AnhTP		
Date	24/10/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Manager 		
Summary:	<ul style="list-style-type: none"> - This use case allows manager to delete a website. 		
Goal:	<ul style="list-style-type: none"> - Website chosen is deleted from manager. 		

Triggers:

- Manager send delete website command.

Preconditions:

- User must login into the system with role Manager.

Post conditions:

- Success: Website is removed by manager.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Manager goes to manage website view.	Display a table show list of website.
2	Manager send delete website command.	System start removing website. [Exception 1] System show message remove successful.

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Removed fail.	Show message to notify manager that remove failed.

Relationships: WCS_UC_1.

Business Rules:

- In case of success scenarios, website is removed in the system.

Table 35 - <Manager> Delete website use case specification

2.3.3.6 Create new staff

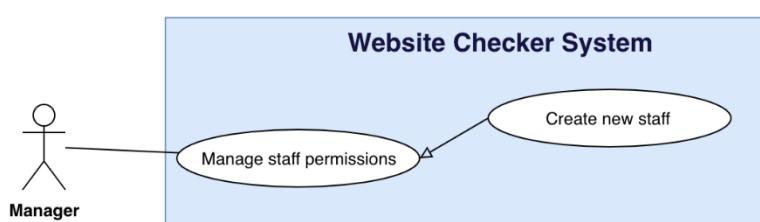


Figure 35 - <Manager> Create new staff

USE CASE - WCS_UC_30			
Use Case No.	WCS_UC_30	Use Case Version	1.0

Use Case Name	Create new staff		
Author	AnhTP		
Date	24/10/2018	Priority	Normal

Actor:

- Manager

Summary:

- This use case allows manager to create a new staff.

Goal:

- Manager successfully creates new staff.

Triggers:

- Manager send create new staff command to the system.

Preconditions:

- User must login into the system with role Manager.

Post conditions:

- Success: New staff successful store in database.
- Fail: New website cannot be saved.

Main Success Scenario:

Step	Actor Action	System Response
1	Manager goes to create new staff screen.	Create new staff screen is shown with following labels and fields: <ul style="list-style-type: none"> • Name staff: text, required, length 8-30. • Username staff: text, required, length 8-30. • Password: text, required, length 8-30.
2	Manager inputs information.	
3	Manager send create new staff command.	Create new staff in the system. [Exception 1] System shows message staff has been created successfully.

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Create new staff fail.	Show message to notify manager that create new staff failed.

Relationships: WCS_UC_1.**Business Rules:**

- In case of success scenarios, a new staff is created in the system.
- New staff username is not duplicate.

Table 36 - <Manager> Create new staff use case specification

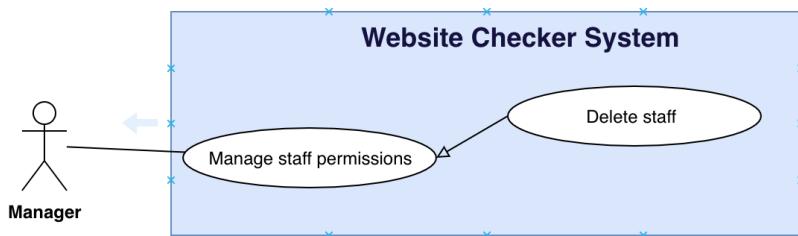
2.3.3.7 Delete Staff

Figure 36 - <Manager> Delete staff

USE CASE – WCS_UC_31

Use Case No.	WCS_UC_31	Use Case Version	1.0
Use Case Name	Delete staff		
Author	AnhTP		
Date	24/10/2018	Priority	Normal

Actor:

- Manager

Summary:

- This use case allows manager to delete staff.

Goal:

- Manager successfully delete staff in the system.

Triggers:

- Manager send delete command to the system.

Preconditions:

- User must login into the system with role Manager.

Post conditions:

- Success: Staff is removed by manager.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Manager goes to manage staff screen.	Display a table show list of staffs.

2	<p>Manager send delete staff command.</p>	<p>System start removing website. [Exception 1] System show message remove successful</p>
---	---	---

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Removed fail.	Show message to notify manager that remove failed.

Relationships: WCS_UC_1.**Business Rules:**

- In case of success scenarios, a staff is deleted.
- Staff is deleted who cannot log in the system.

Table 37 - <Manager> Delete staff use case specification

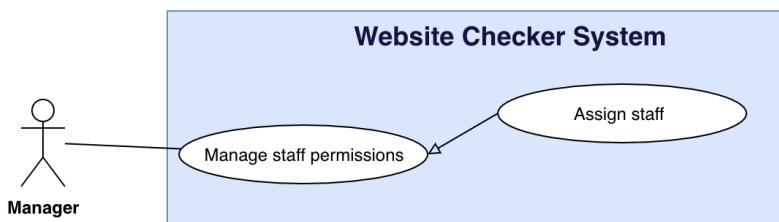
2.3.3.8 Assign Staff

Figure 37 <Manager> Assign staff

USE CASE - WCS_UC_32			
Use Case No.	WCS_UC_32	Use Case Version	1.0
Use Case Name	Assign staff		
Author	AnhTP		
Date	24/10/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Manager 		
Summary:	<ul style="list-style-type: none"> - This use case allows manager to assign staff. 		
Goal:	<ul style="list-style-type: none"> - Manager successfully assign staff website in the system. 		
Triggers:			

- Manager send assign staff command to the system.

Preconditions:

- User must login into the system with role Manager.

Post conditions:

- Success: Staff is assigned website by manager.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Manager goes to assign staff screen.	Assign staff screen is shown with following labels and fields: - List staff: dropdown list.
2	Manager chooses staff.	
2	Manager send assign staff command.	Assign staff website in the system. [Exception 1] System shows message staff has been assigned successfully.

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Assign fail.	Show message to notify manager that remove failed.

Relationships: WCS_UC_1.**Business Rules:**

- In case of success scenarios, a staff is assigned.
- Staff can log in and test website assigned.

Table 38 <Manager> Assign staff use case specification

2.3.3.9 Add new word website's dictionary

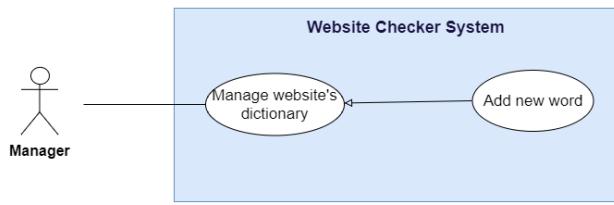


Figure 38 <Manager> Add new word website's dictionary

USE CASE – WCS_UC_33															
Use Case No.	WCS_UC_33	Use Case Version	1.0												
Use Case Name	Add new word														
Author	TrinhND														
Date	24/10/2018	Priority	Normal												
Actor:	<ul style="list-style-type: none"> - Manager 														
Summary:	<ul style="list-style-type: none"> - Allow Manager add new word to the website's dictionary. 														
Goal:	<ul style="list-style-type: none"> - Manager can add new word to the website's dictionary. 														
Triggers:	<ul style="list-style-type: none"> - Manager sends add new word to website's dictionary command. 														
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Manager. 														
Post conditions:	<ul style="list-style-type: none"> - Success: A new word will be added to the website's dictionary. - Fail: System shows error messages. 														
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Manager goes to Manage Website's Dictionary view.</td><td>System render Manage Website's Dictionary screen.</td></tr> <tr> <td>2</td><td>Manager sends add new word command.</td><td> Manage Website's Dictionary screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. • Word form: text, required. • Definition: text, required. </td></tr> <tr> <td>3</td><td>Manager fills data. [Alternative 1],[Exception 1,2,3,4]</td><td>Display Manage Website's Dictionary view with word list table include new word.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Manager goes to Manage Website's Dictionary view.	System render Manage Website's Dictionary screen.	2	Manager sends add new word command.	Manage Website's Dictionary screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. • Word form: text, required. • Definition: text, required. 	3	Manager fills data. [Alternative 1],[Exception 1,2,3,4]	Display Manage Website's Dictionary view with word list table include new word.
Step	Actor Action	System Response													
1	Manager goes to Manage Website's Dictionary view.	System render Manage Website's Dictionary screen.													
2	Manager sends add new word command.	Manage Website's Dictionary screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. • Word form: text, required. • Definition: text, required. 													
3	Manager fills data. [Alternative 1],[Exception 1,2,3,4]	Display Manage Website's Dictionary view with word list table include new word.													

Alternative Scenario:

No	Actor Action	System Response
1	Manager clicks the “Cancel” button.	Display Manage Website’s Dictionary view.

Exceptions:

No	Actor Action	System Response
1	Manager add a word has been existed.	Show popup “This word is existed”.
2	Word input is null.	“Done” button will be disabled.
3	Word form input is null.	“Done” button will be disabled.
4	Definition input is null.	“Done” button will be disabled.

Relationships: WCS_UC_1, WCS_UC_7.**Business Rules:**

- Website’s dictionary will be uses for test spelling.
- Manager can add new word to Website’s Dictionary base on the content will be checked.

Table 39 <Manager> Add new word website's dictionary use case specification

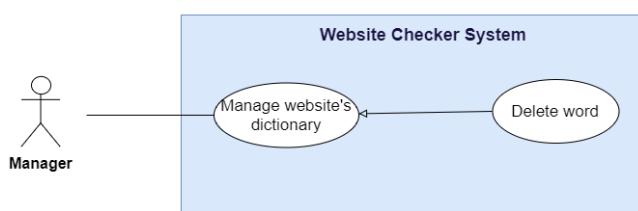
2.3.3.10 Delete word website's dictionary

Figure 39 <Manager> Delete word website's dictionary

USE CASE – WCS_UC_34			
Use Case No.	WCS_UC_34	Use Case Version	1.0
Use Case Name	Delete word		
Author	TrinhND		
Date	24/10/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Manager 		

Summary:

- Allow Manager delete word from the website's dictionary.

Goal:

- A word will be deleted from the website's dictionary.

Triggers:

- Manager sends delete word command.

Preconditions:

- User must login into the system with role Manager.

Post conditions:

- Success: A word will be deleted from the website's dictionary.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Manager goes to Manage Website's Dictionary view.	System render Manage Website's Dictionary screen.
2	Manager sends delete word command.	System show popup to confirm the command.
3	Manager confirm to delete word. [Alternative 1]	Display Manage Website's Dictionary view with word list table not include word has been deleted.

Alternative Scenario:

No	Actor Action	System Response
1	Manager clicks the "No" button.	Display Manage Website's Dictionary view.

Exceptions: N/A**Relationships:** WCS_UC_1, WCS_UC_9.**Business Rules:**

- Website's dictionary will be uses for test spelling.
- A word has been deleted will not be used for the next test.

Table 40<Manager> Delete word website's dictionary use case specification

2.3.3.11 Add the warning word list

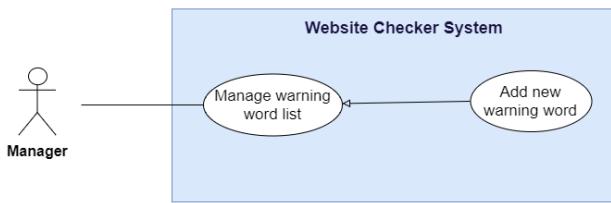


Figure 40 <Manager> Add new warning word

USE CASE – WCS_UC_35															
Use Case No.	WCS_UC_35	Use Case Version	1.0												
Use Case Name	Add new warning word														
Author	TrinhND														
Date	24/10/2018	Priority	Normal												
Actor:	<ul style="list-style-type: none"> - Manager 														
Summary:	<ul style="list-style-type: none"> - Allow Manager add new word to the warning word list. 														
Goal:	<ul style="list-style-type: none"> - Manager can add new word to the warning word list. 														
Triggers:	<ul style="list-style-type: none"> - Manager sends add new warning word command. 														
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Manager. 														
Post conditions:	<ul style="list-style-type: none"> - Success: A new word will be added to the warning word list. - Fail: System shows error messages. 														
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Manager goes to Manage Wordlist view.</td><td>System render Manage Wordlist screen.</td></tr> <tr> <td>2</td><td>Manager sends add new warning word command.</td><td>Manage Wordlist screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. Type: text, required. </td></tr> <tr> <td>3</td><td>Manager fills data. [Alternative 1],[Exception 1,2]</td><td>Display Manage Wordlist view with word list table include new word.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Manager goes to Manage Wordlist view.	System render Manage Wordlist screen.	2	Manager sends add new warning word command.	Manage Wordlist screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. Type: text, required.	3	Manager fills data. [Alternative 1],[Exception 1,2]	Display Manage Wordlist view with word list table include new word.
Step	Actor Action	System Response													
1	Manager goes to Manage Wordlist view.	System render Manage Wordlist screen.													
2	Manager sends add new warning word command.	Manage Wordlist screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. Type: text, required.													
3	Manager fills data. [Alternative 1],[Exception 1,2]	Display Manage Wordlist view with word list table include new word.													
Alternative Scenario:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> </table>			No	Actor Action	System Response									
No	Actor Action	System Response													

1	Manager cancel add new warning word command.	Display Manage Wordlist view.
---	--	-------------------------------

Exceptions:

No	Actor Action	System Response
1	Manager add a word has been existed.	Show popup "This word is existed".
2	Word input is null.	"Done" button will be disabled.

Relationships: WCS_UC_1, WCS_UC_9.**Business Rules:**

- Manager can choose type of warning word base on topic.
- Warning word list will be uses for test prohibited content.
- Manager can add new word to warning word list base on the content will be checked.

Table 41 <Manager> Add new warning word use case specification

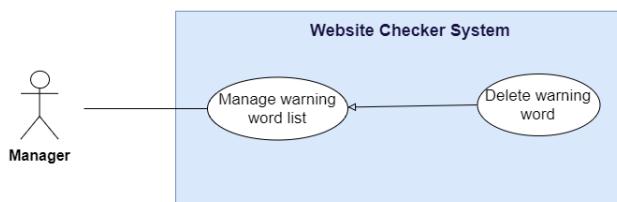
2.3.3.12 Delete warning word

Figure 41 <Manager> Delete warning word

USE CASE - WCS_UC_36			
Use Case No.	WCS_UC_36	Use Case Version	1.0
Use Case Name	Delete warning word		
Author	TrinhND		
Date	24/10/2018	Priority	Normal
Actor:			
- Manager			
Summary:			
- Allow Manager delete word from the warning word list.			
Goal:			
- A word will be deleted from the warning word list.			

Triggers:

- Manager sends delete warning word command.

Preconditions:

- User must login into the system with role Manager.

Post conditions:

- Success: A word will be deleted from the warning word list.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Manager goes to Manage Wordlist view.	System render Manage Wordlist screen.
2	Manager sends delete word command.	System show popup to confirm the command.
3	Manager confirm to delete word. [Alternative 1]	Display Manage Wordlist view with word list table not include word has been deleted.

Alternative Scenario:

No	Actor Action	System Response
1	Manager cancel delete word command.	Display Manage Wordlist view.

Exceptions: N/A**Relationships:** WCS_UC_1, WCS_UC_9.**Business Rules:**

- Warning word list will be uses for test prohibited content.
- A word has been deleted will not be used for the next test.

Table 42 <Manager> Delete warning word use case specification

2.3.3.13 Edit warning word

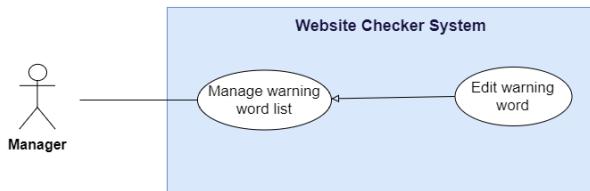


Figure 42 <Manager> Edit warning word

USE CASE – WCS_UC_37															
Use Case No.	WCS_UC_37	Use Case Version	1.0												
Use Case Name	Edit warning word														
Author	TrinhND														
Date	24/10/2018	Priority	Normal												
Actor:	<ul style="list-style-type: none"> - Manager 														
Summary:	<ul style="list-style-type: none"> - Allow Manager edit word of the warning word list. 														
Goal:	<ul style="list-style-type: none"> - Manager can edit word of the warning word list. 														
Triggers:	<ul style="list-style-type: none"> - Manager sends edit warning word command. 														
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Manager. 														
Post conditions:	<ul style="list-style-type: none"> - Success: Data of word will be updated and add to the warning word list. - Fail: System shows error messages. 														
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Manager goes to Manage Wordlist view.</td><td>System render Manage Wordlist screen.</td></tr> <tr> <td>2</td><td>Manager sends edit word command.</td><td> Manage Wordlist screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. Type: text, required. </td></tr> <tr> <td>3</td><td>Manager fills data. [Alternative 1], [Exception 1]</td><td>Display Manage Wordlist view with word list table include new data of word.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Manager goes to Manage Wordlist view.	System render Manage Wordlist screen.	2	Manager sends edit word command.	Manage Wordlist screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. Type: text, required.	3	Manager fills data. [Alternative 1], [Exception 1]	Display Manage Wordlist view with word list table include new data of word.
Step	Actor Action	System Response													
1	Manager goes to Manage Wordlist view.	System render Manage Wordlist screen.													
2	Manager sends edit word command.	Manage Wordlist screen is shown with following labels and fields: <ul style="list-style-type: none"> • Word: text, required. Type: text, required.													
3	Manager fills data. [Alternative 1], [Exception 1]	Display Manage Wordlist view with word list table include new data of word.													
Alternative Scenario:															

No	Actor Action	System Response
1	Manager clicks the “Cancel” button.	Display Manage Wordlist view.

Exceptions:

No	Actor Action	System Response
1	Word input is null.	“Done” button will be disabled.

Relationships: WCS_UC_1, WCS_UC_9.**Business Rules:**

- Manager can choose type of warning word base on topic.
- Newest data of the warning word list will be uses for test prohibited content.

Table 43 <Manager> Edit warning word use case specification

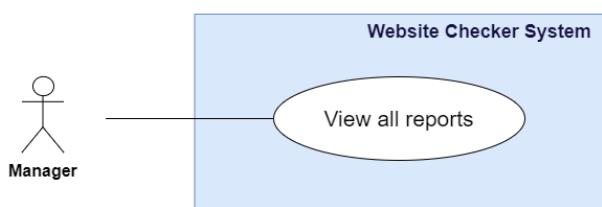
2.3.3.14 View all reports

Figure 43 <Manager> View all reports

USE CASE - WCS_UC_37			
Use Case No.	WCS_UC_37	Use Case Version	1.0
Use Case Name	View All Reports		
Author	TrinhND		
Date	24/10/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Manager 		
Summary:	<ul style="list-style-type: none"> - Allow Manager to view all reports have been created. 		
Goal:	<ul style="list-style-type: none"> - Manager can view all reports have been created. 		
Triggers:	<ul style="list-style-type: none"> - Manager sends view all reports command. 		
Preconditions:			

- User must login into the system with role Manager.

Post conditions:

- Success: all reports have been created will be show by created time.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Manager sends view all reports command.	System render view all reports screen. [Exception 1]
2	Staff send view detail report by Staff command.	Show details report created by Staff.

Alternative Scenario: N/A

Exceptions:

No	Actor Action	System Response
1	There are not report has been created.	History report table is empty.

Relationships: WCS_UC_1, WCS_UC_7 ... to WCS_UC_22.

Business Rules:

- At least one report should be done and save.
- History report will be display by Staff and filter by time created.
- Manager can view all report by each Staff.

Table 44 <Manager> View all reports use case specification

3. Software System Attribute

3.1 Usability

Website's UI is fit for each browser on each device

- Font style: Arial, Sans-serif
- Font size: 12px -26px
- Color: green, black, red, white, blue, Light Slate Grey, Fuego, Honeysuckle...
- Background: White, Catalina Blue, Whisper...

3.2 Reliability

- The information storing on the database is permanent.

3.3 Availability

- The system must be available at any time 24/7.

3.4 Security

- Only Manager has fully access to the system
- Input data is validated before saving to database.

3.5 Maintainability

- The system is divided into separated modules for easy maintain.

3.6 Portability

- Web application can be run on Chrome browser version 42 or later.

3.7 Performance

- System has successfully test with basic standard

4. Conceptual Diagram

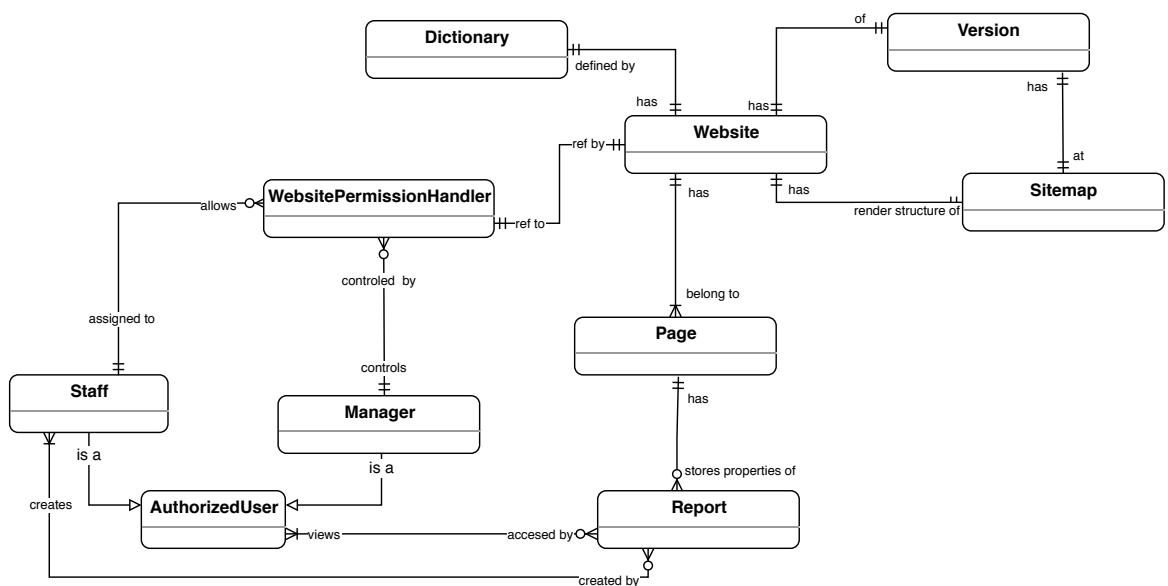


Figure 44 Conceptual Diagram

Data Dictionary

Entity Data dictionary: describe all content of all entities	
Entity Name	Description
Authorized User	Abstract entity describes a user in system

Staff	Contain the staff information
Manager	Contain the manager information
Permission Handler	Represent a staff is assigned to maintain each page
Website	Contain the website information
Dictionary	Contain the dictionary information
Page	Contain the page information
Sitemap	Contain the sitemap information
Report	Contain the page information

Table 45 Conceptual Diagram Data Dictionary

D. Software Design Description

1. Design Overview

- This document describes the technical and user interface design of MSSC System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
- The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
- The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
- The database design describes the relationships between entities and details of each entity.
- Document overview:

Section 2: gives an overall description of the system architecture design.

Section 3: gives component diagrams that describe the connection and integration of the system.

Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.

Section 5: describe screens design.

Section 6: describe a fully attributed ERD.

Section 7: describe algorithms

2. System Architectural Design

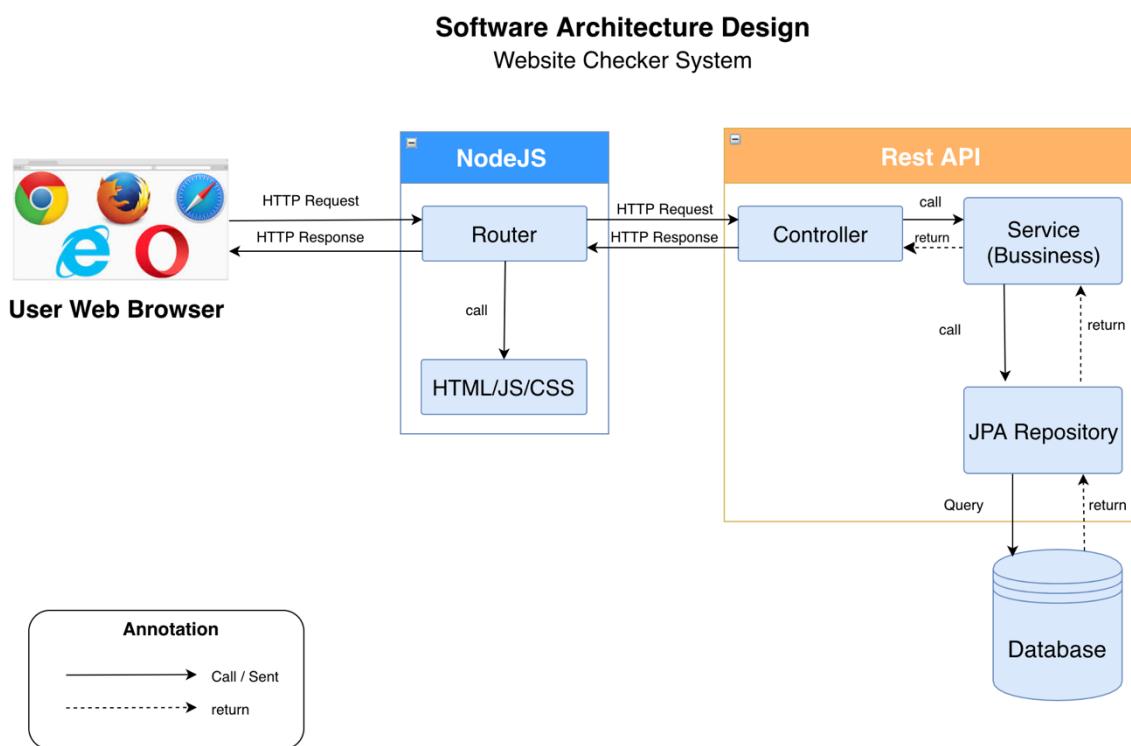


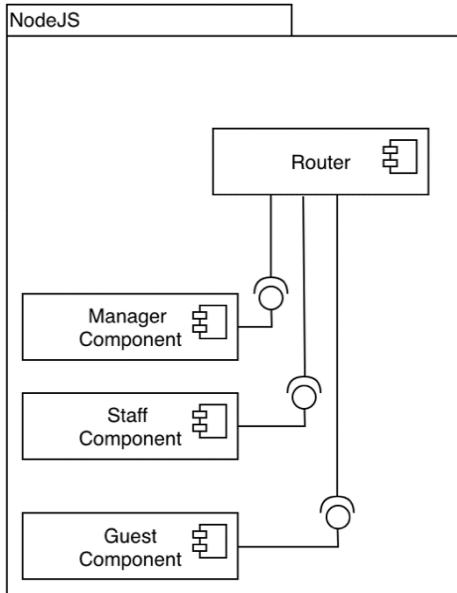
Figure 45 Software Architecture Design

In Web Application, the system is developed under MVC architecture style. We choose this architecture for Web Application because of following advantages:

- Ecosystem: Java has a big ecosystem. It has a lot of libraries and support for nearly every platform (OS, Database, etc.).
- It offers wide support for enterprise capabilities like messaging (Using JMS), Webservices, Rest and big data.
- ReactJS is quite easy to be mastered.
- ReactJS is easy to maintain isolated components in React.
- Constant re-rendering of components provides efficient arranging at increasing complexity.

- Convenient React is highly competitive to MVC. One-way data flow provides maintainability and efficient arrangement of data and DOM elements.
- It is the most light-weighted framework among the ones that are widely used today.

3. Component Diagram

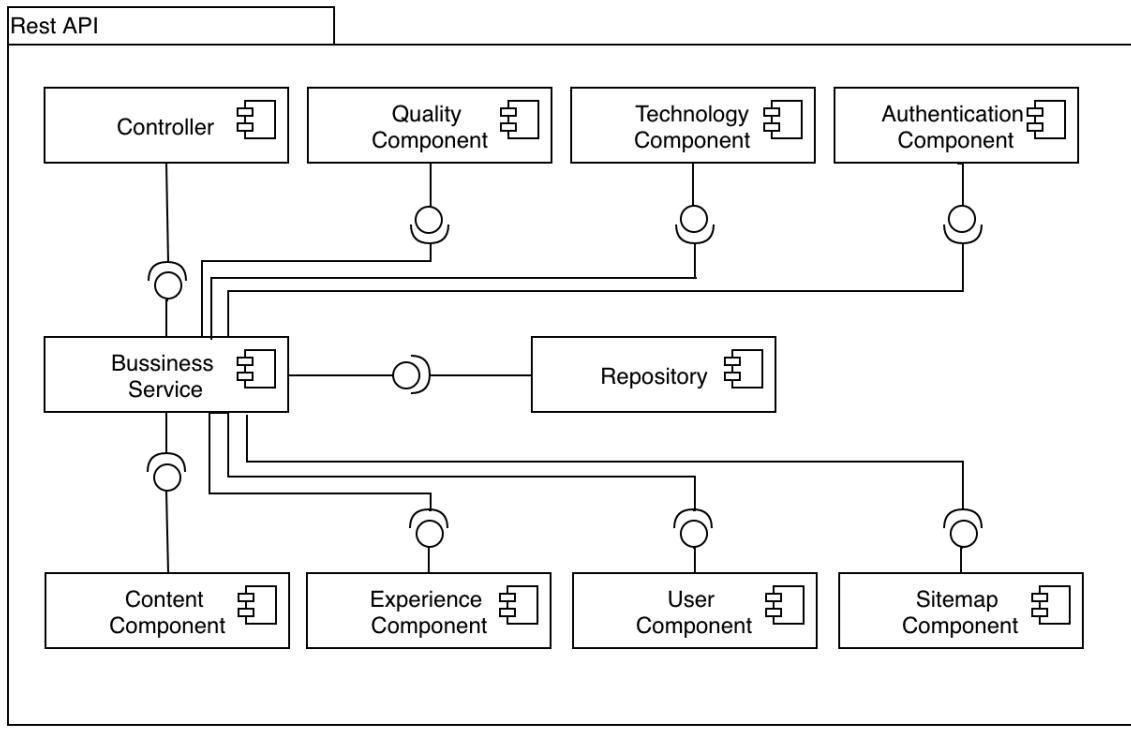


Front End Component Diagram

Figure 46 Front End Component Diagram

COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
NodeJS	
Router	Handle request and response, accept input and convert it to commands for back end.
Manager Component	Handle manager activities in the system
Staff Component	Handle staff activities in the system
Guest Component	Handle guest activities in the system

Table 46 Front End Component Diagram Dictionary



Back End Component Diagram

Figure 47 Back End Component Diagram

COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
Rest API	
Controller	Handle requests and responses; accept input and convert it to commands for model and view.
Bussiness Servies	Handle system's business operations
Repository	Store Data

Table 47 Back End Component Diagram Dictionary

4. Detailed Description

4.1 Class Diagram

Class diagram

Website Checker System

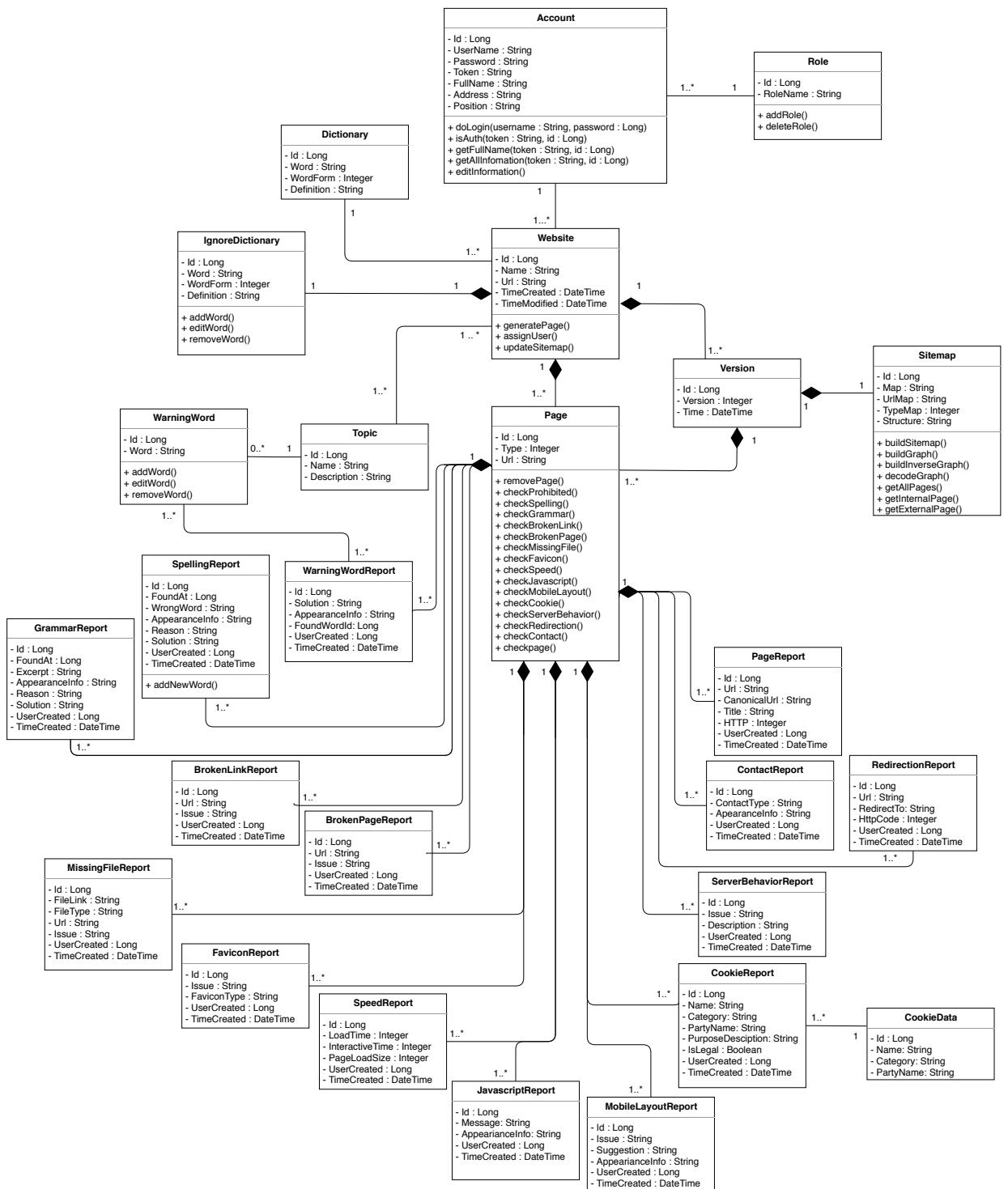


Figure 48 Class Diagram

4.2 Class Diagram Explanation

CLASS DICTIONARY: DESCRIBE CLASS		
Class Name	Mapping column with Conceptual diagram	Description
Account	N/A	Not exist in conceptual diagram but need this table to contain the user information
Role	N/A	Not exist in conceptual diagram but need this table to contain the information of a role
Website	Website	Contains the website information
Dictionary	Dictionary	Contains the dictionary information
Sitemap	Sitemap	Contains the sitemap information
Page	Page	Contains the page information
Topic	N/A	Not exist in conceptual diagram but need this table to contain the information of a topic information
Warning Word	N/A	Not exist in conceptual diagram but need this table to contain the information of a warning word information
Version	Version	Contains the information of version information
GrammarReport	Report	Contains the grammar report information
SpellingReport	Report	Contains the spelling report information
WarningWorReport	Report	Contains the warning word report information
PageReport	Report	Contains the page report information

BrokenLinkReport	Report	Contains the broken link report information
BrokenPageReport	Report	Contains the broken page report information
ContactReport	Report	Contains the contact report information
RedirectionReport	Report	Contains the redirection report information
MissingFileReport	Report	Contains the missing file report information
FaviconReport	Report	Contains the favicon report information
SpeedReport	Report	Contains the speed report information
JavascriptReport	Report	Contains the javascript error report information
MobileLayoutReport	Report	Contains the mobile layout report information
CookieReport	Report	Contains the cookie report information
ServerBehaviorReport	Report	Contains the server behavior report information

Table 48 Class Diagram Dictionary

4.1.1 Account

Attribute	Type	Visibility	Description
Id	String	Private	Unique identifier of an account
Username	String	Private	Account's username
Password	String	Private	Account's password

Fullscreen	String	Private	Account's fullname
Address	String	Private	Account's address
Position	String	Private	Account's position

Table 49 <Class Diagram Attributes> Account

Method	Redirect Type	Visibility	Description
doLogin	Object	Public	Login
isAuth	Object	Public	Authentication
getFullName	Object	Public	Get full name
getAllInformation	Object	Public	Get all information
editInformation	Object	Public	Edit information

Table 50 <Class Diagram Methods> Account

4.1.2 Role

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a role
Rolename	String	Private	Role's name

Table 51 <Class Diagram Attributes> Role

Method	Redirect Type	Visibility	Description
addRole	boolean	Public	Add new role
deleteRole	boolean	Public	Delete role

Table 52 <Class Diagram Methods> Role

4.1.3 Website

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a website

Name	String	Private	Website's name
Url	String	Private	Website's url
TimeCreated	Datetime	Private	Website's time created
TimeModified	Datetime	Private	Website's time modified

Table 53 <Class Diagram Attributes> Website

Method	Redirect Type	Visibility	Description
generatePage	object	Public	Generate page
assignUser	object	Public	Assign user
updateSitemap	object	Public	Update sitemap

Table 54 <Class Diagram Methods> Website

4.1.4 Dictionary

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a word
Word	String	Private	Word
WordForm	Integer	Private	Word Form
Definition	String	Private	Word's definition

Table 55 <Class Diagram Attributes> Dictionary

Method	Redirect Type	Visibility	Description
addWord	Void	Public	Add new word
editWord	Void	Public	Edit word
removeWord	Void	Public	Remove word

Table 56 <Class Diagram Methods> Dictionary

4.1.5 Sitemap

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a sitemap
Map	String	Private	Sitemap's map
Urlmap	String	Private	Url root
TypeMap	Integer	Private	Type map

Table 57 <Class Diagram Attributes> Sitemap

Method	Redirect Type	Visibility	Description
buildSitemap	Void	Public	Build sitemap
getAllPage	List	Public	Get all page from sitemap
getInternalPages	List	Public	Get all internal pages

Table 58 <Class Diagram Methods> Sitemap

4.1.6 WarningWord

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a warning word
Word	String	Private	Warning word

Table 59 <Class Diagram Attributes> Warningword

Method	Redirect Type	Visibility	Description
addWord	Boolean	Public	Add new warning word
editWord	Boolean	Public	Edit warning word
removeWord	Boolean	Public	Remove warning word

Table 60 <Class Diagram Methods> Warningword

4.1.7 Topic

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a topic
Name	String	Private	Topic's name
Description	String	Private	Topic's description

Table 61 <Class Diagram Attributes> Topic

Method	Redirect Type	Visibility	Description
addTopic	boolean	Public	Add new topic
editTopic	boolean	Public	Edit topic
removeTopic	Boolean	Public	Remove topic

Table 62 <Class Diagram Methods> Topic

4.1.8 Page

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a page
Type	Integer	Private	Page's name
Url	String	Private	Page's url

Table 63 <Class Diagram Attributes> Page

Method	Redirect Type	Visibility	Description
Getter	Attribute Type	Public	Get attribute value
Setter	Void	Public	Set attribute value
removePage	Void	Public	Remove page
checkProhibited	List	Public	Check prohibited word in page
checkSpelling	List	Public	Check spelling word on page
checkGrammar	List	Public	Check grammar on page

checkBrokenLink	List	Public	Check broken link on page
checkBrokenPage	List	Public	Check broken page on page
checkMissingFile	List	Public	Check missing file on page
checkFavicon	List	Public	Check favion on page
checkSpeed	List	Public	Measure the speed of page
checkMobileLayout	List	Public	Check page's mobile layout
checkJavascript	List	Public	Check javascript error on page
checkCookie	List	Public	Check cookie on page
checkServerBehavior	List	Public	Check Server Behavior
checkRedirection	List	Public	Check page's redirection
checkContact	List	Public	Check contact detail on page
checkPage	List	Public	Check page information

Table 64 <Class Diagram Methods> Page

4.1.9 Version

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a version
Version	Integer	Private	Version
Time	Datetime	Private	Time created

Table 65 <Class Diagram Attributes> Version

4.1.10 GrammarReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a grammar report

FoundAt	Long	Private	Grammar error is found at position
Excerpt	String	Private	Grammar errors excerpt
AppereanceInfo	String	Private	Appereance info
Reason	String	Private	Grammar error reason
Solution	String	Private	Grammar error solution
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 66 <Class Diagram Attributes> GrammarReport

4.1.11 SpellingReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a spelling report
FoundAt	Long	Private	Spelling error is found at
WrongWord	String	Private	Wrong word
AppereanceInfo	String	Private	Appereance info
Reason	String	Private	Spelling error reason
Solution	String	Private	Spelling error solution
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 67 <Class Diagram Attributes> SpellingReport

Method	Redirect Type	Visibility	Description
addNewWord	boolean	Public	Add new word

Table 68 <Class Diagram Methods> SpellingReport

4.1.12 WarningWordReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a warning word report
Solution	String	Private	Warning word solution
AppereanceInfo	String	Private	Appereance info
FoundWordId	Long	Private	Found word id
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 69 <Class Diagram Attributes> WarningWordReport

4.1.13 PageReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a page report
Url	String	Private	Page's url
CanonicalUrl	String	Private	Page's canonical url
Tittle	String	Private	Page's title
HTTP	Integer	Private	Page's http code response
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 70 <Class Diagram Attributes> PageReport

4.1.14 ContactReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a contact detail report

ContactType	String	Private	Email or phone number
AppereanceInfo	String	Private	Page's url has Contact detail
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 71 <Class Diagram Attributes> ContactReport

4.1.15 BrokenLinkReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a broken link report
Url	String	Private	Broken link's url
Issue	String	Private	Broken link's issue
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 72 <Class Diagram Attributes> BrokenLinkReport

4.1.16 RedirectionReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a redirection report
Url	String	Private	Page's url is redirection
RedirectTo	String	Private	url is redirected
HttpCode	Integer	Private	Http code response
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 73 <Class Diagram Attributes> RedirectionReport

4.1.17 BrokenPageReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a broken page report
Url	String	Private	Broken page's url
Issue	String	Private	Broken page's issue
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 74 <Class Diagram Attributes> BrokenPageReport

4.1.18 MissingFileReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a missing file report
FileLink	String	Private	File's link
FileType	String	Private	File's type
Url	String	Private	Page's url has missing file
Issues	String	Private	Missing file issues
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 75 <Class Diagram Attributes> MissingFileReport

4.1.19 FaviconReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a favicon report
Issue	String	Private	File's link

FaviconType	String	Private	Favicon's type
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 76 <Class Diagram Attributes> FaviconReport

4.1.20 SpeedReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a speed report
LoadTime	Integer	Private	Time to load
InteractiveTime	Integer	Private	Time to interactive
PageLoadSize	Integer	Private	Page load size
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 77 <Class Diagram Attributes> SpeedReport

4.1.21 JavascriptReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a javascript error report
MessageInfo	String	Private	Javascript error message
AppereanceInfo	String	Private	Appereance info
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 78 <Class Diagram Attributes> JavascriptReport

4.1.22 MobileLayoutReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a mobile layout report
Issues	String	Private	Mobile layout issues
Suggestion	String	Private	Mobile layout suggestion
AppereanceInfo	String	Private	Appereance info
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 79<Class Diagram Attributes> MobileLayoutReport

4.1.23 CookieReport

Attribute	Type	Visibility	Description
Id	Long	Private	Unique identifier of a cookie report
Name	String	Private	Cookie's name
Category	String	Private	Cookie's category
PartyName	String	Private	Party name has cookie
PurposeDescription	String	Private	Cookie's purpose description
IsLegal	String	Private	Cookie is legal
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 80<Class Diagram Attributes> CookieReport

4.1.24 ServerBehaviorReport

Attribute	Type	Visibility	Description

Id	Long	Private	Unique identifier of a server behavior report
Issue	String	Private	Server behavior's issue
Description	String	Private	Server behavior's description
UserCreated	Long	Private	Staff's id created
TimeCreated	Datetime	Private	Time created

Table 81<Class Diagram Attributes> ServerBehaviorReport

4.3 Interaction Diagram

4.3.1 Login

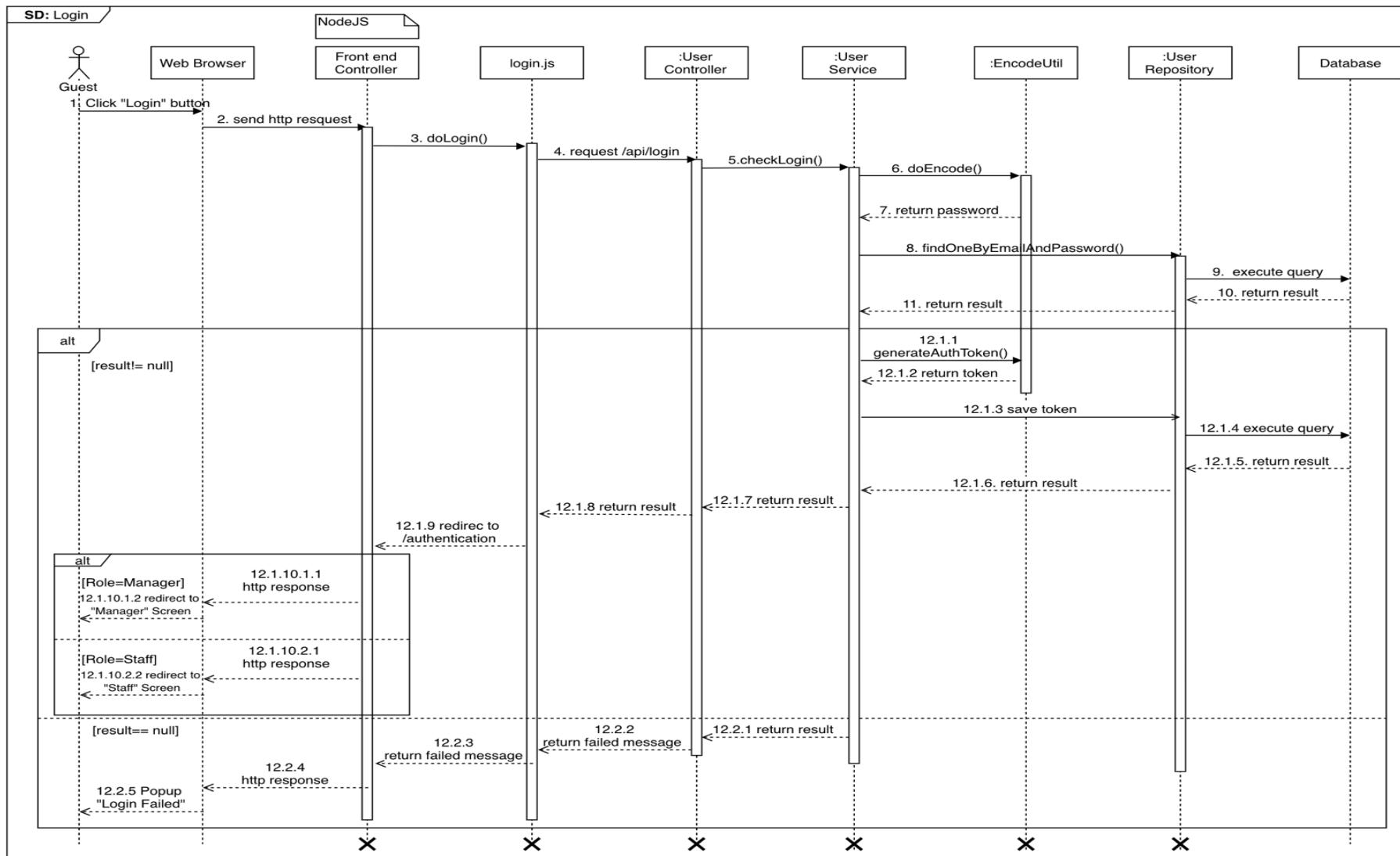


Figure 49 <Sequence Diagram> Guest Login

4.3.2 Test Pages

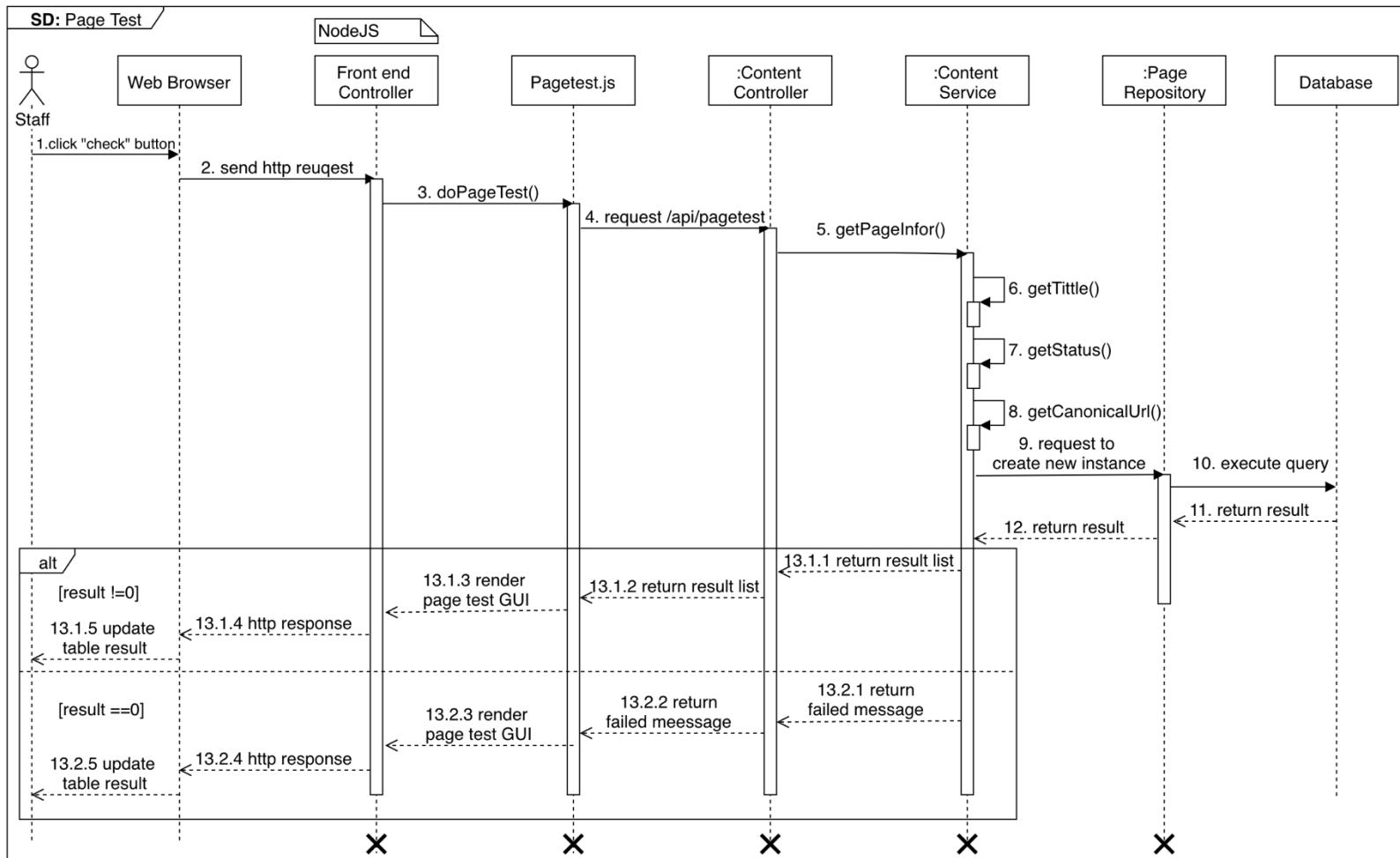


Figure 50 <Sequence Diagram> Staff Test Pages

4.3.3 Test Favicon

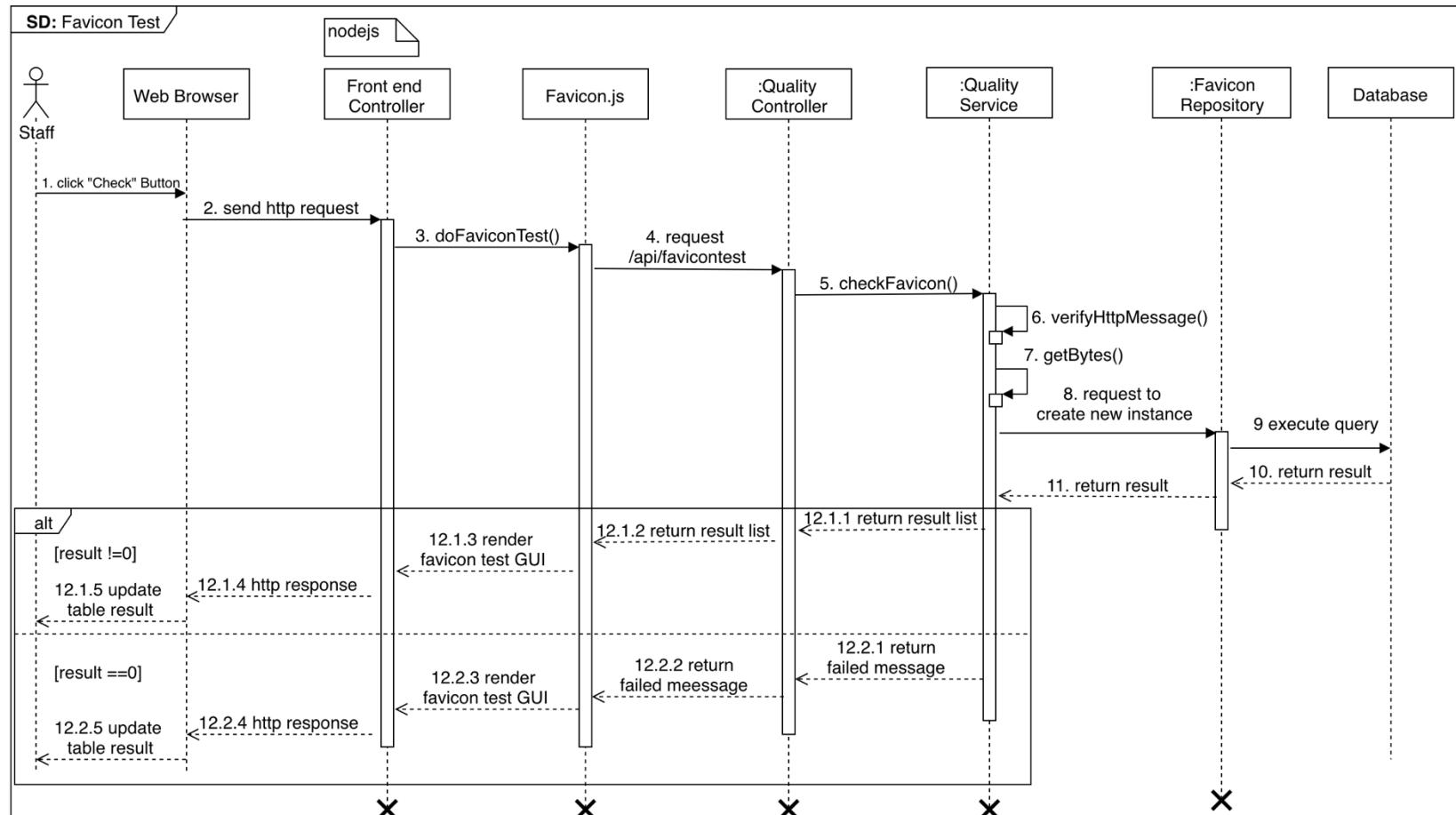


Figure 51 <Sequence Diagram> Staff Test Favicon

4.3.4 Test Contact Details

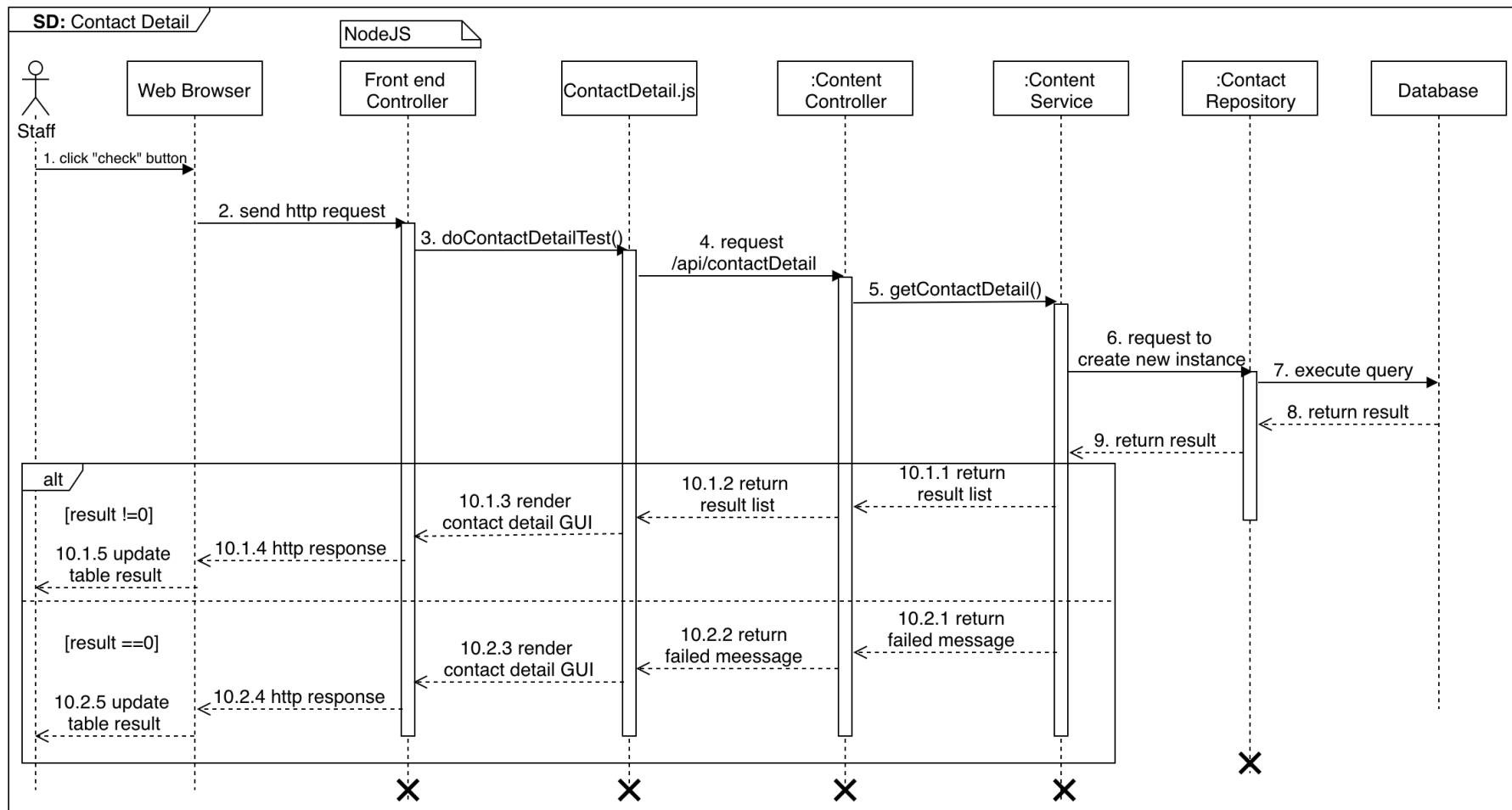


Figure 52<Sequence Diagram> Staff Test Contact Details

4.3.5 Test Redirection

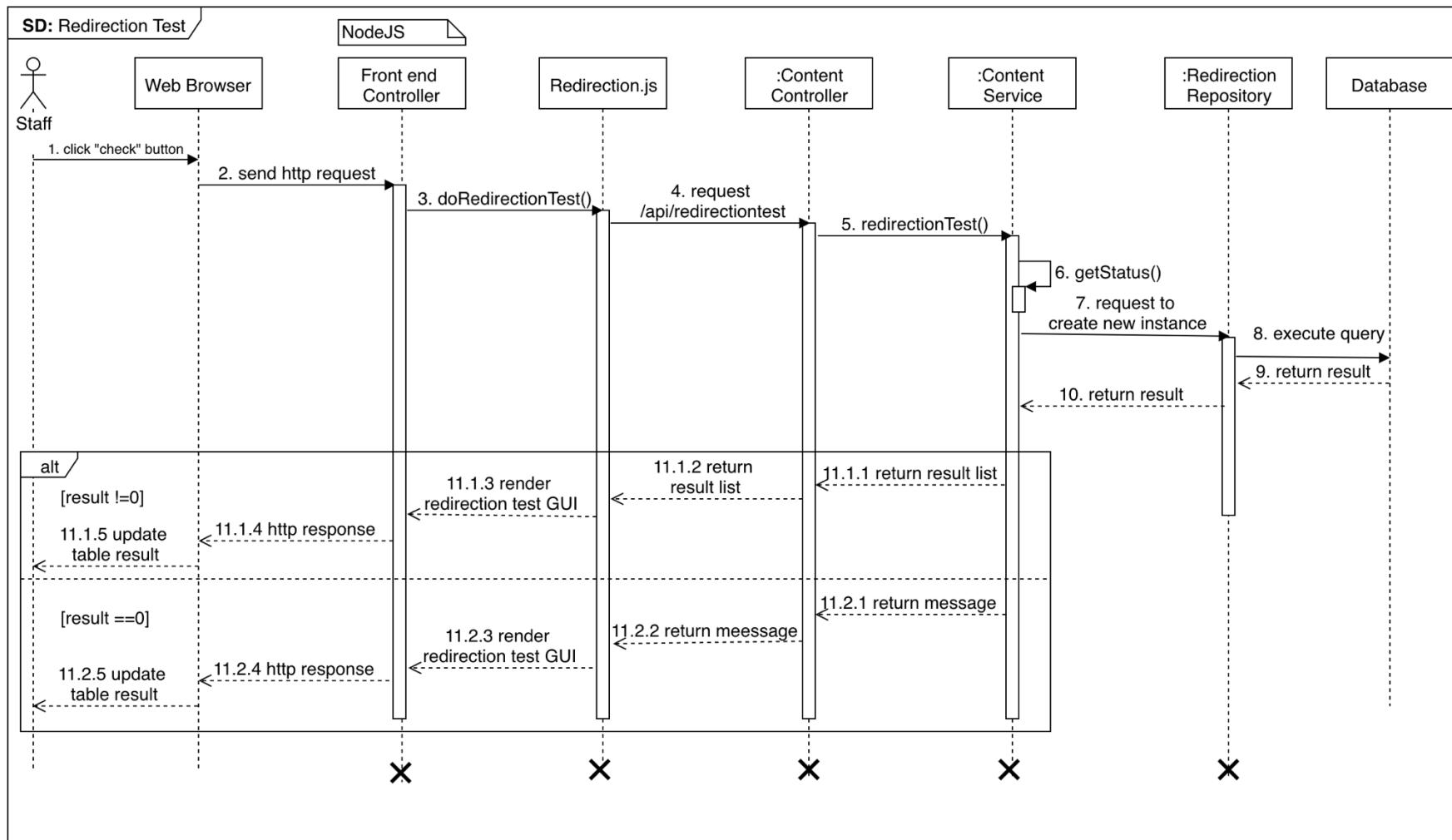
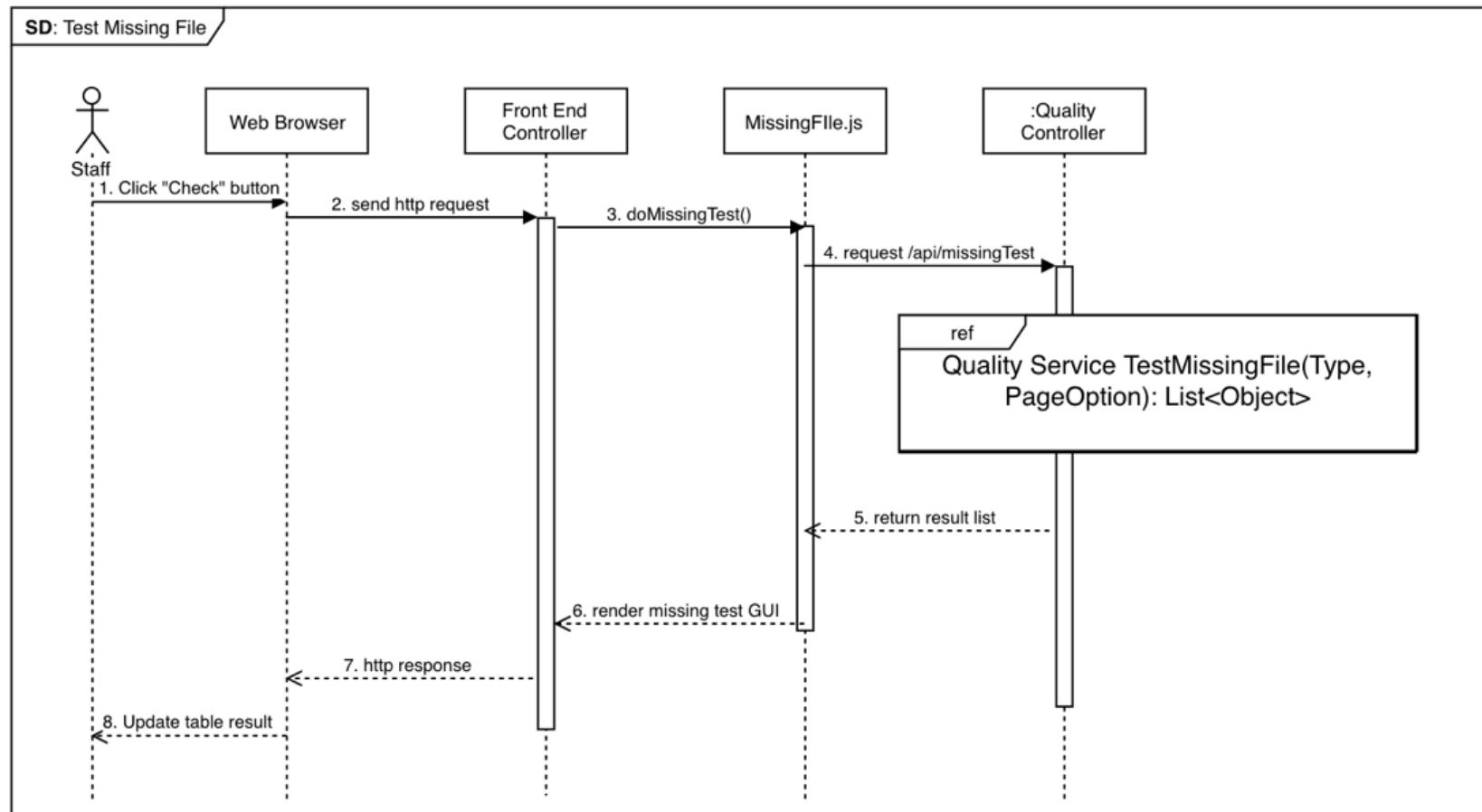


Figure 53 <Sequence Diagram> Staff Test Redirection

4.3.6 Test Missing File



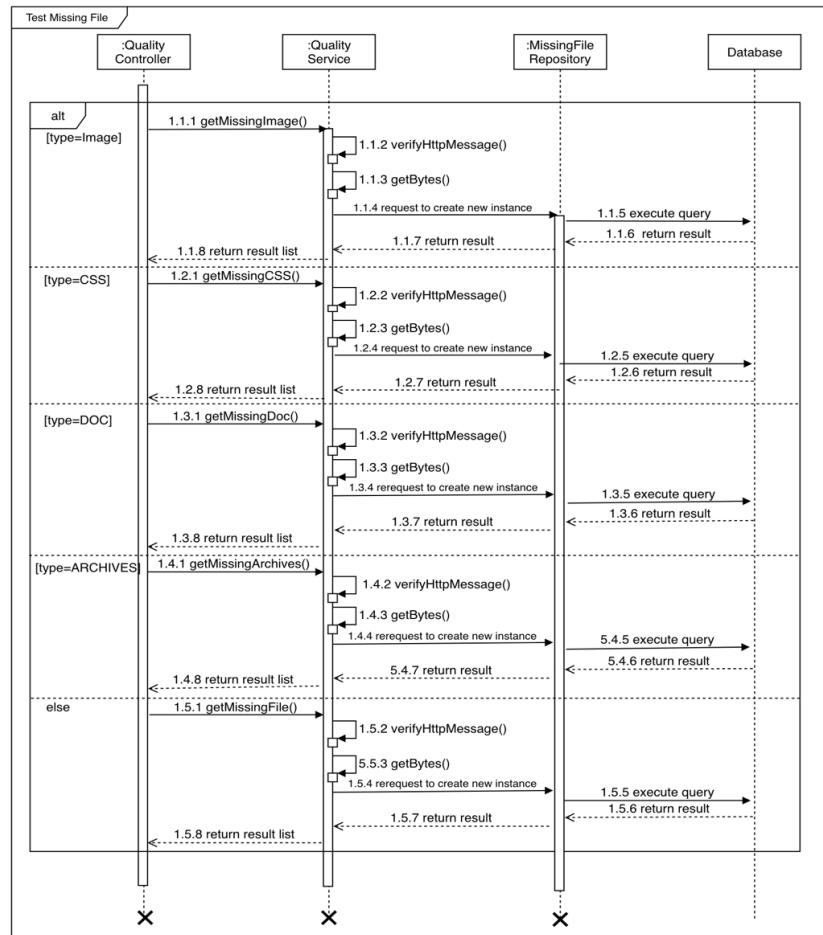


Figure 54 <Sequence Diagram> Staff Test Missing File

4.3.7 Build Sitemap

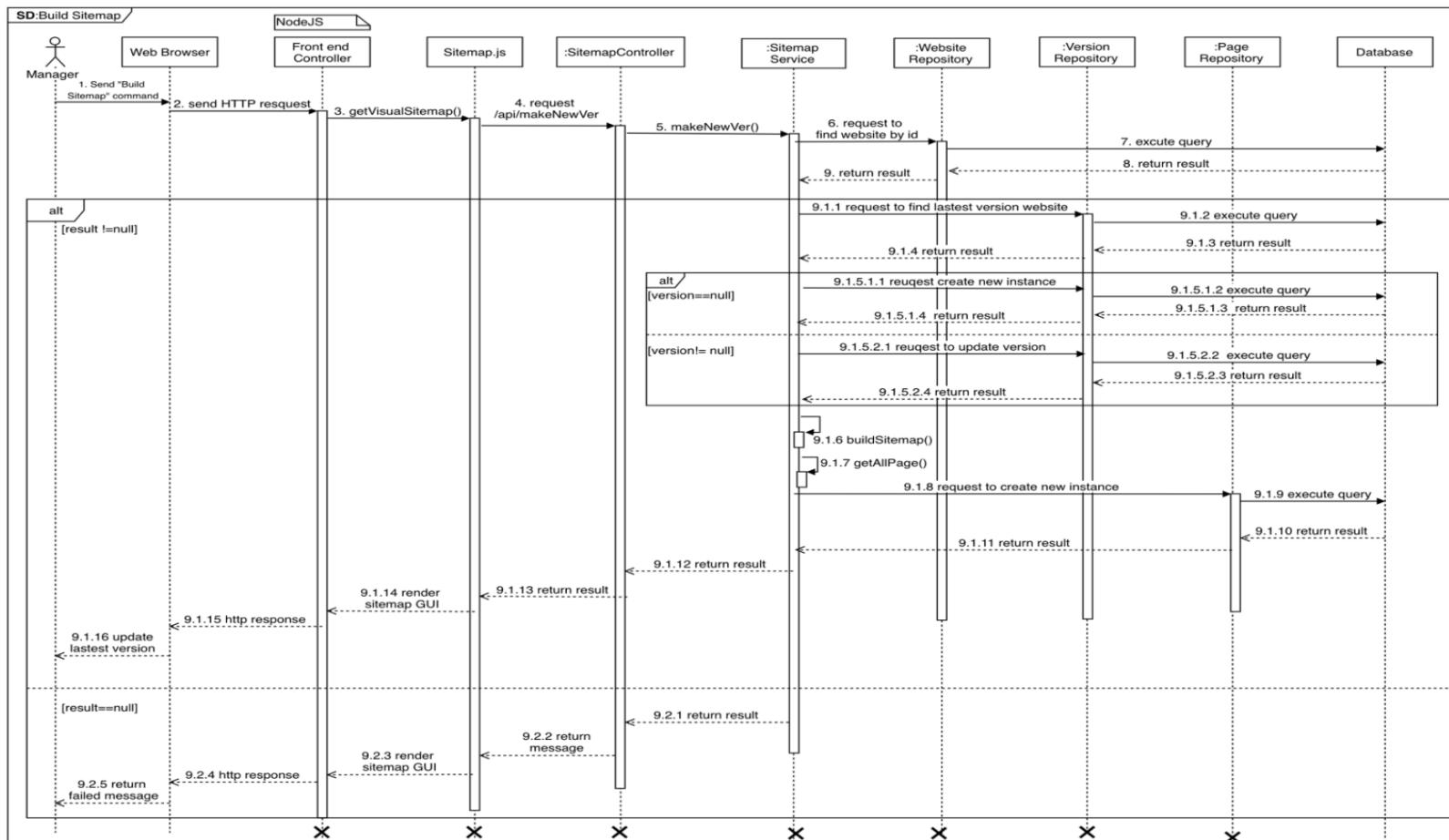


Figure 55 <Sequence Diagram> Manager Build Sitemap

4.3.8 Test Cookie

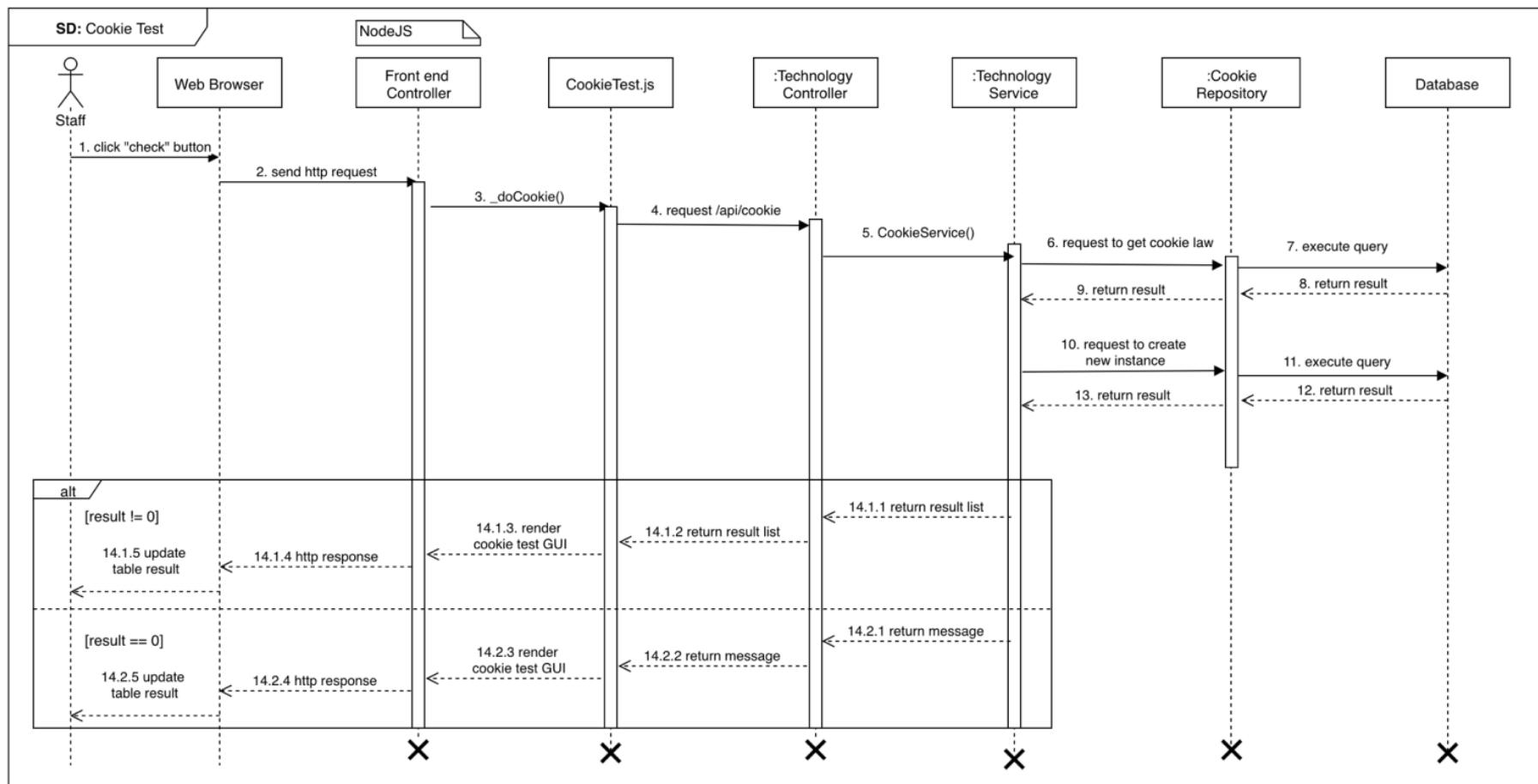


Figure 56 <Sequence Diagram> Staff Test Cookie

4.3.9 Test Broken Page

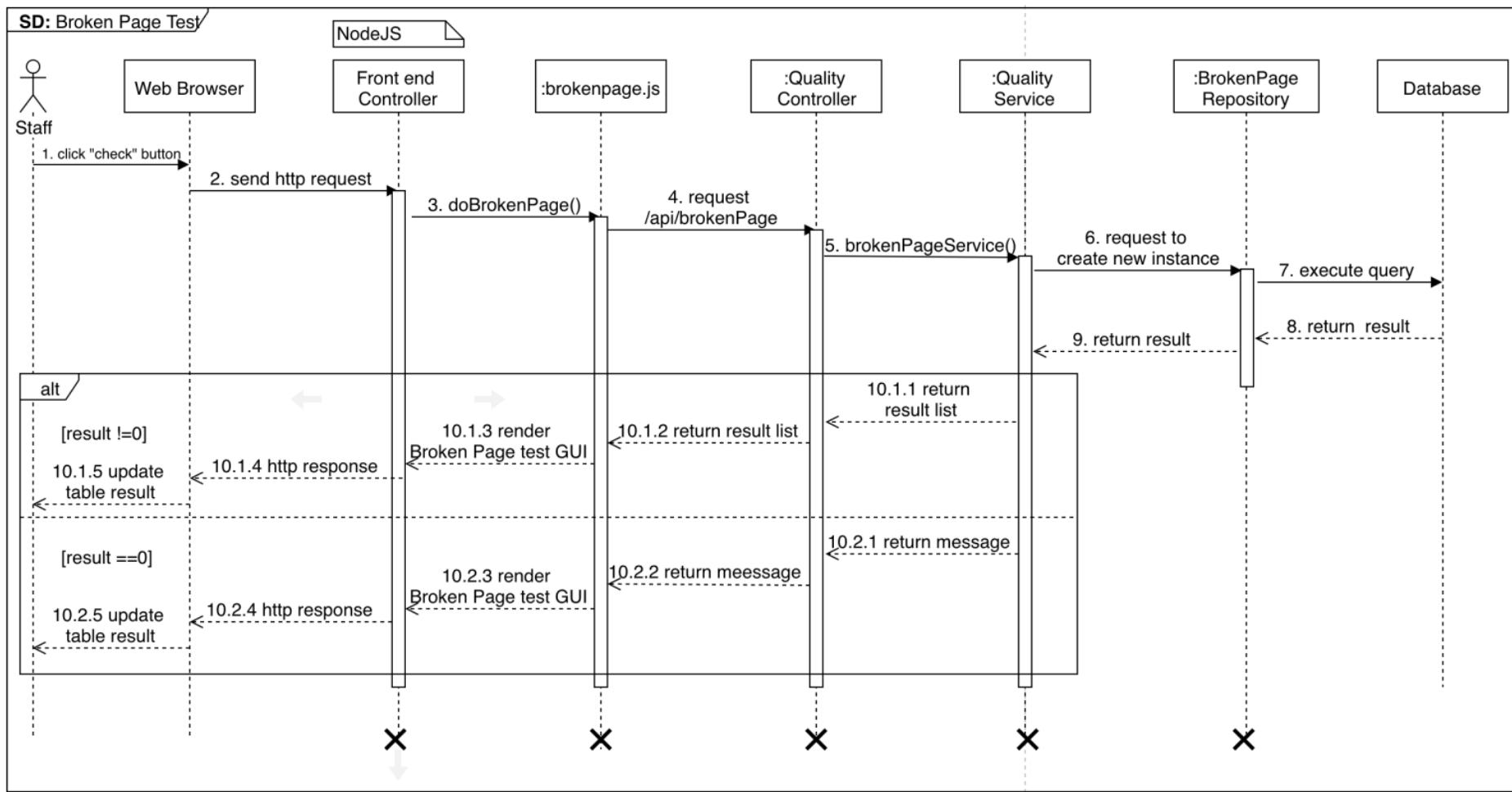


Figure 57 <Sequence Diagram> Staff Test Broken Pages

4.3.10 Test Broken Links

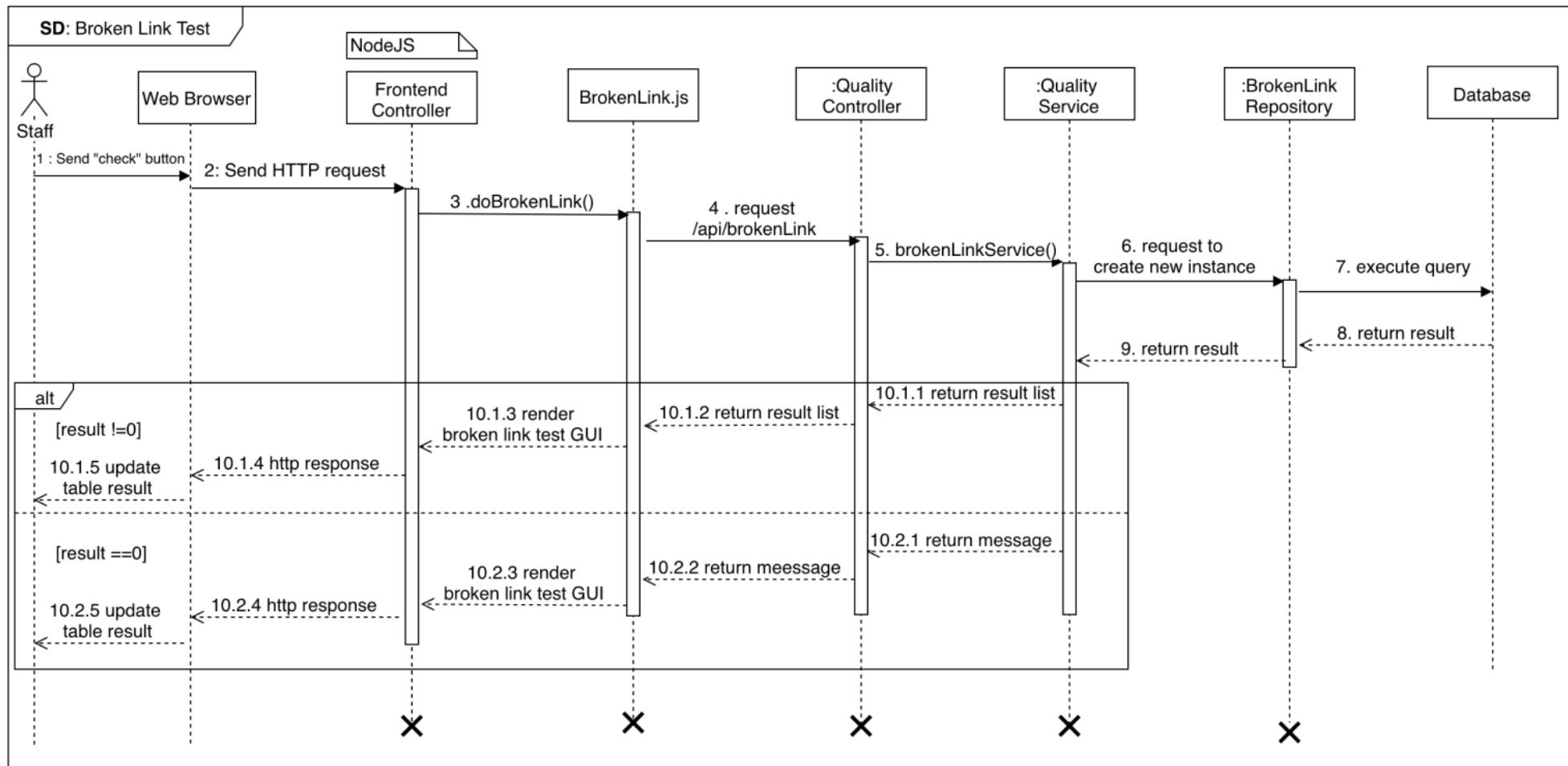


Figure 58 <Sequence Diagram> Staff Test Broken Links

4.3.11 Test Mobile Layout

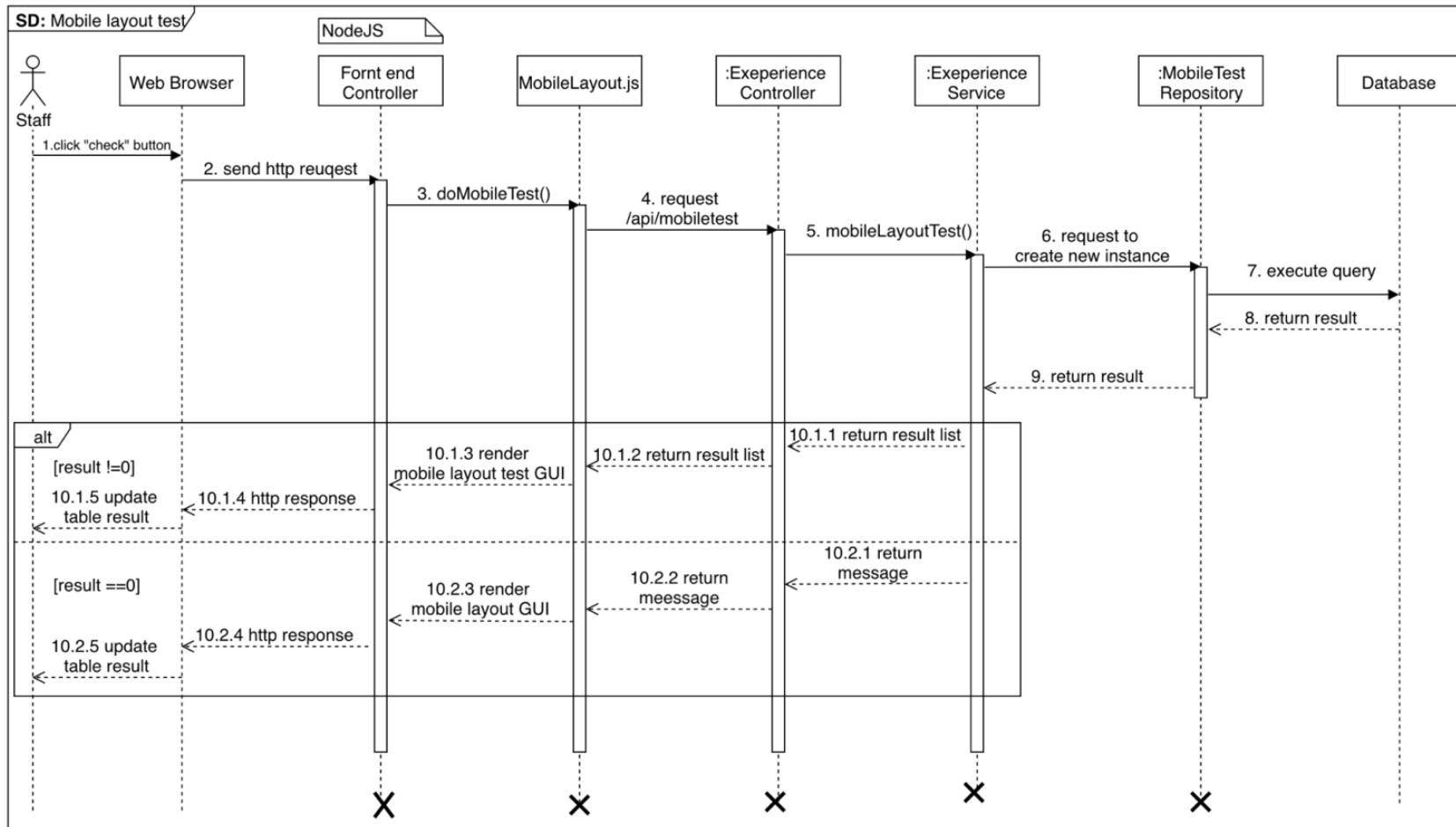


Figure 59 <Sequence Diagram> Staff Test Mobile Layout

4.3.12 Test Javascript errors

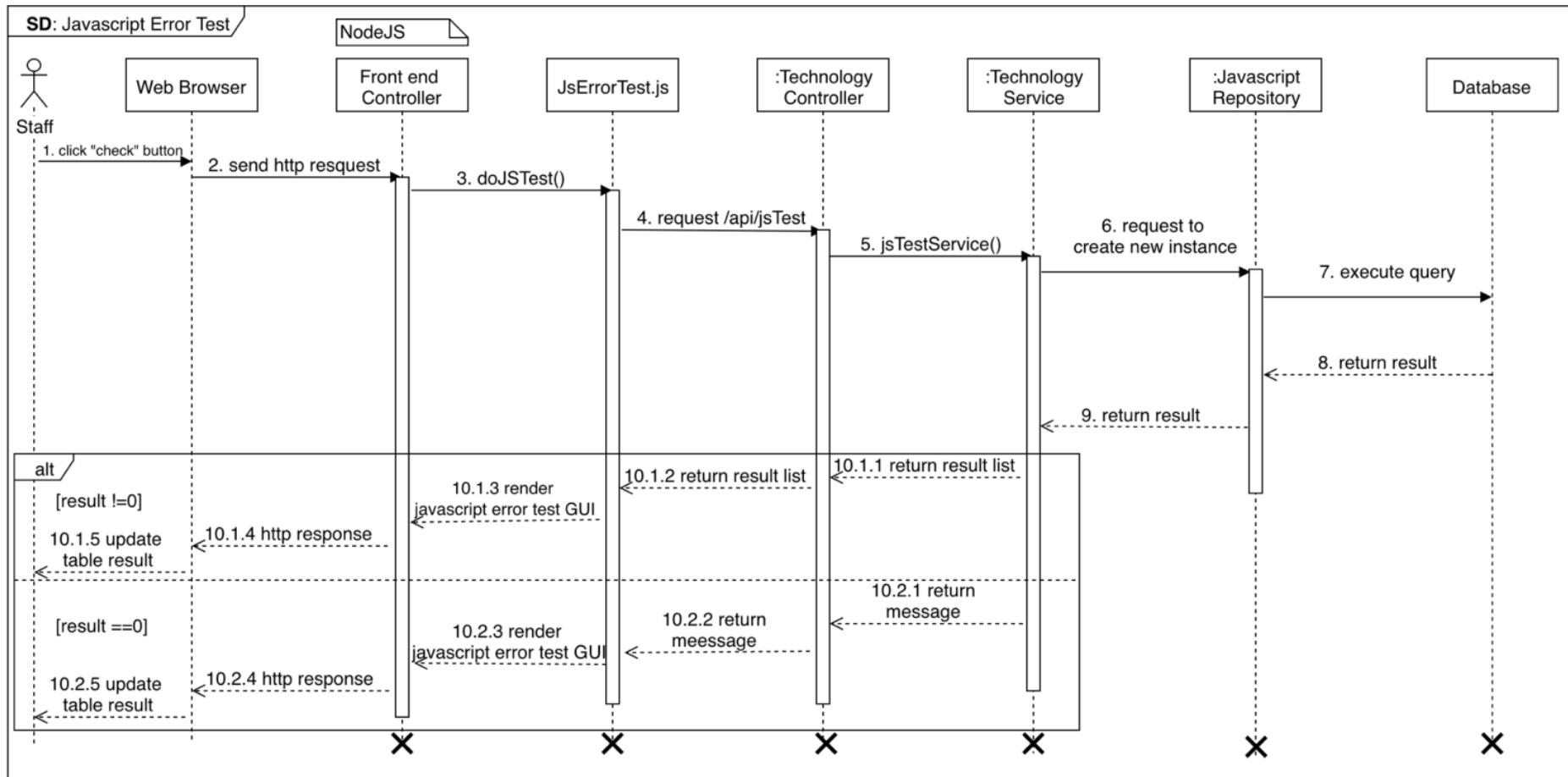


Figure 60 <Sequence Diagram> Staff Test Javascript Errors

4.3.13 Prohibited test

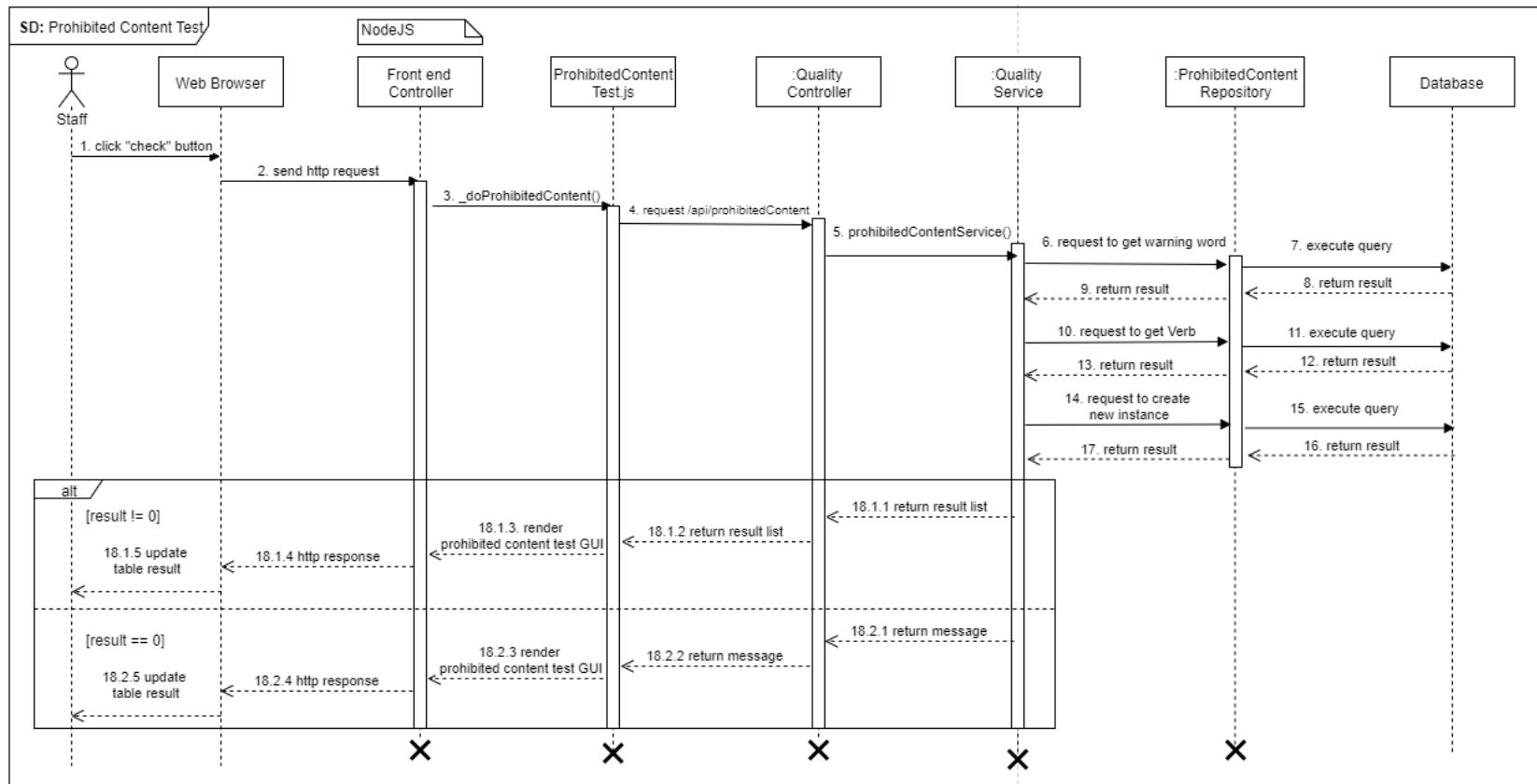


Figure 61 <Sequence Diagram> Staff Test Prohibited

4.3.14 Spelling test

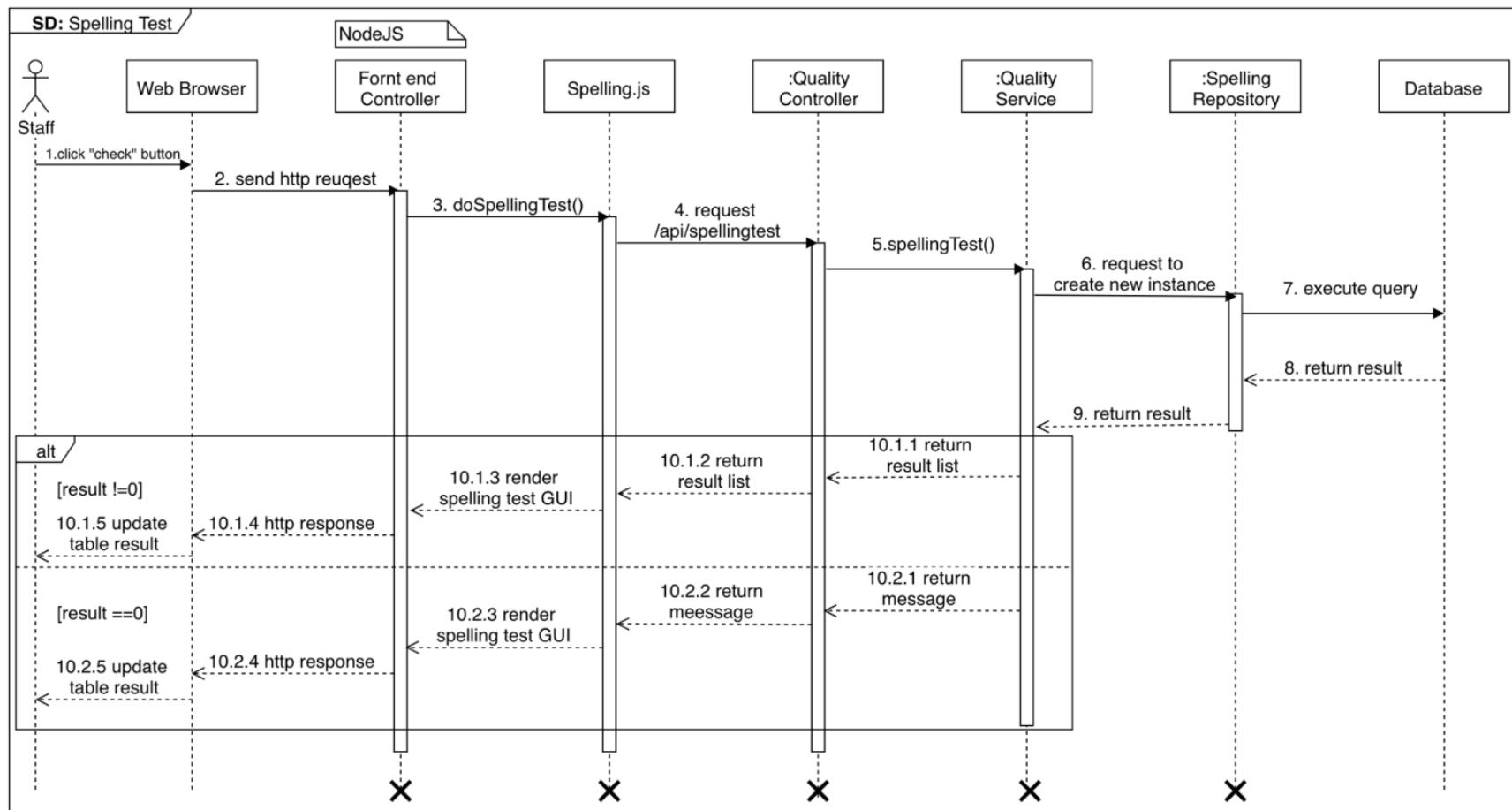


Figure 62 <Sequence Diagram> Staff Test Spelling

4.3.15 Grammar Test

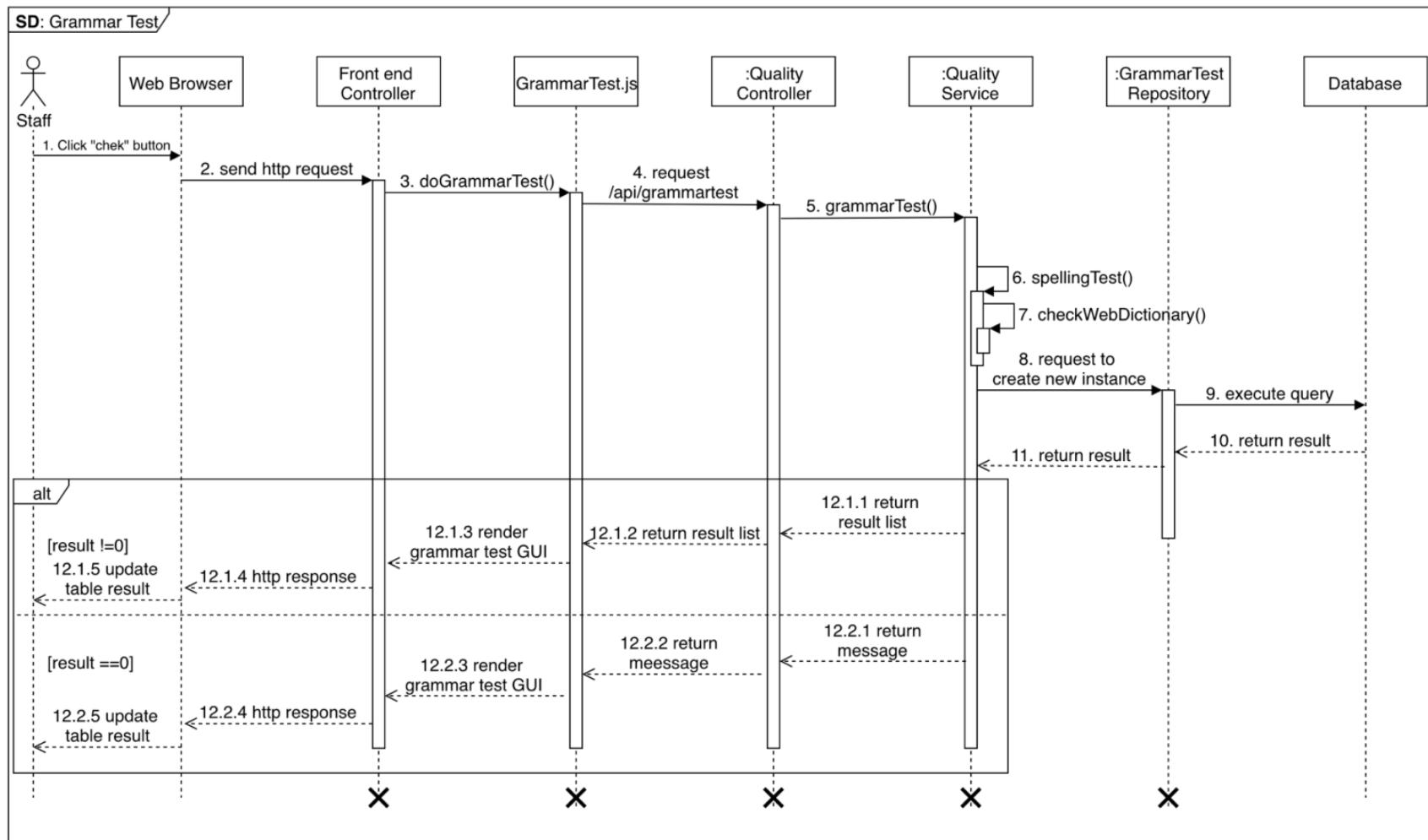


Figure 63 <Sequence Diagram> Staff Test Grammar

4.3.16 Test Server behavior

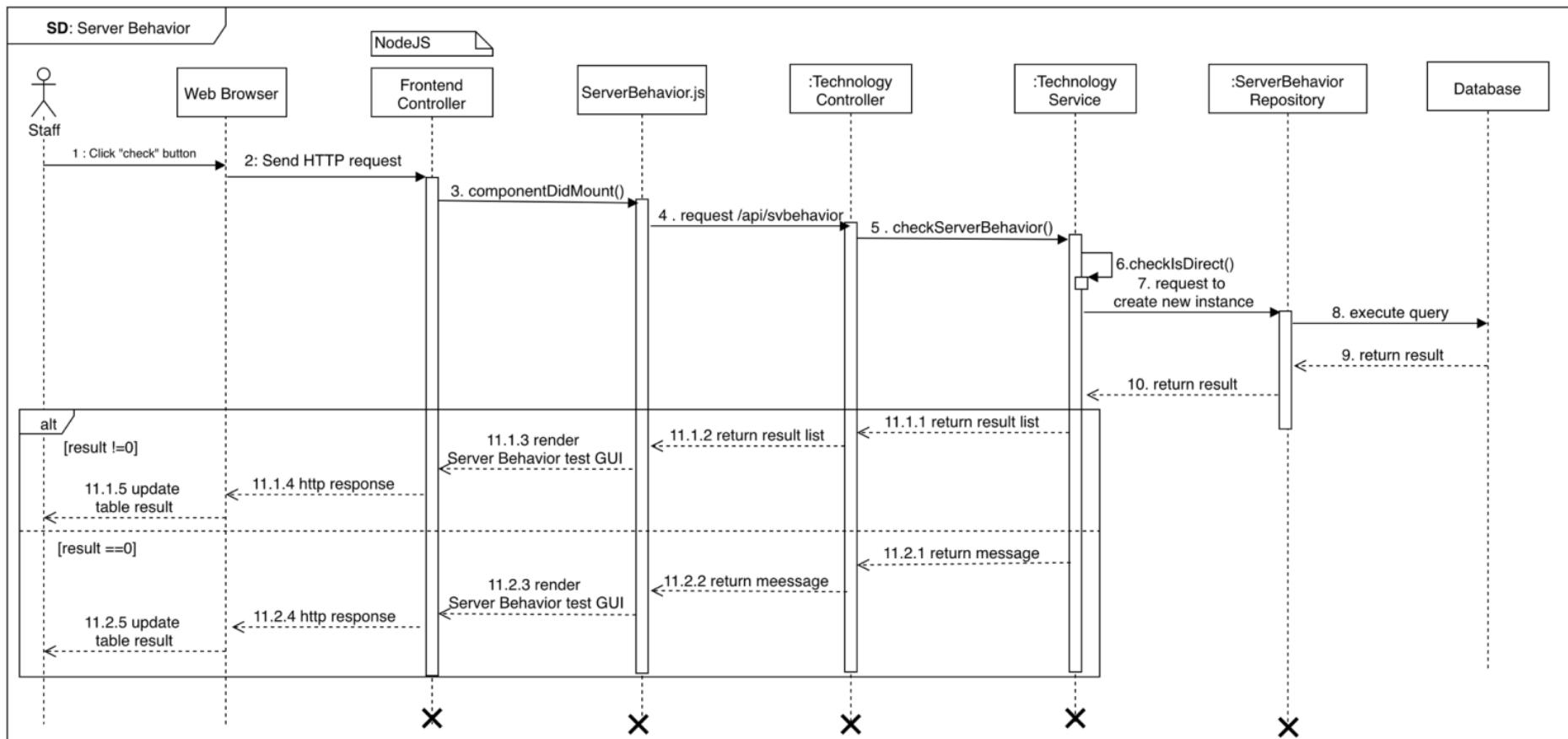


Figure 64 <Sequence Diagram> Staff Test Server Behavior

4.3.17 Measure the speed test

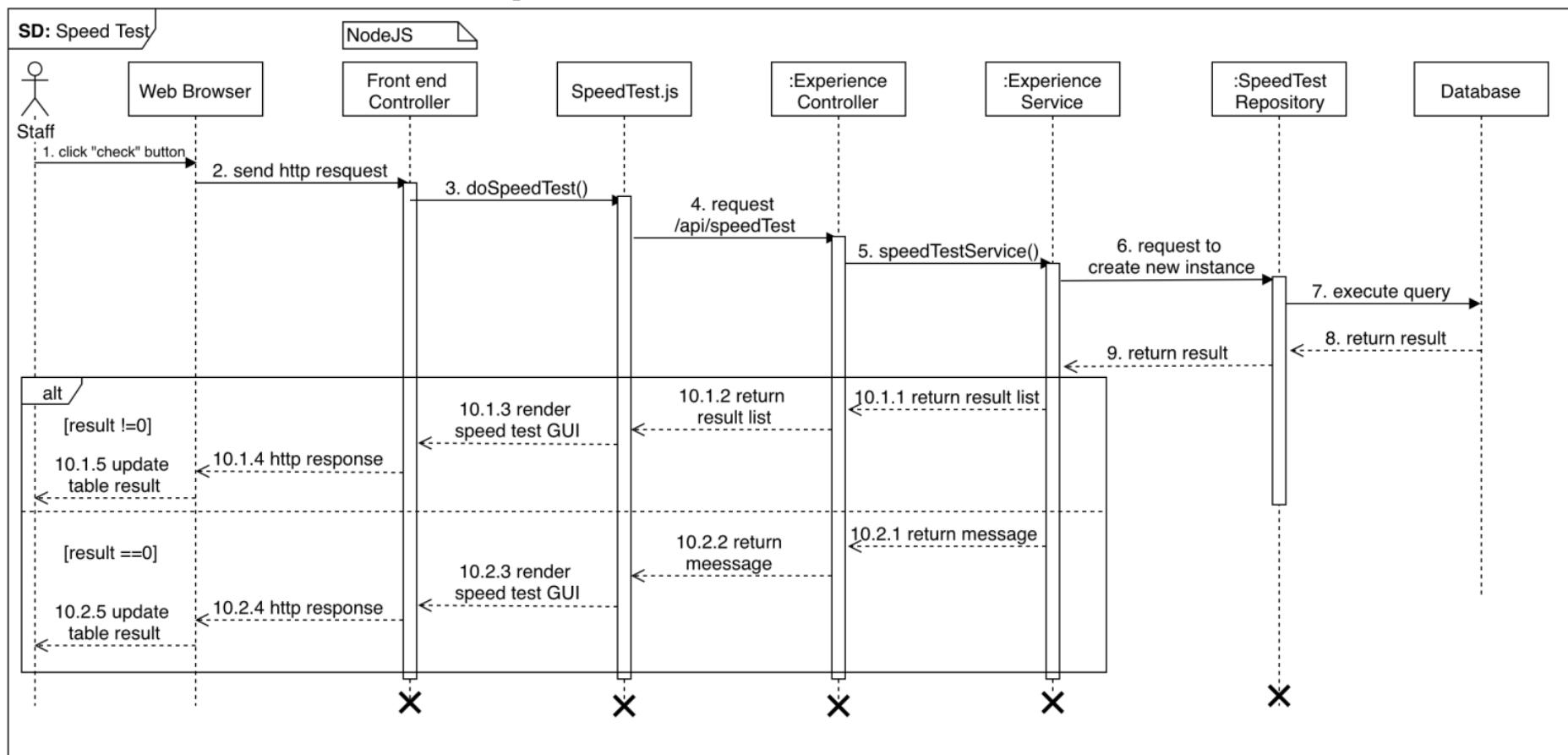


Figure 65 <Sequence Diagram> Staff measure the speed test

5. Interface

5.1 User Interface Design

5.1.1 Login

The screenshot shows a login form with the following elements:

- Field 1:** A yellow input field containing the text "anhtp@gmail.com".
- Field 2:** A yellow input field containing a password represented by dots (".....").
- Field 3:** A blue button labeled "Login".
- Forgot Password:** A link below the login button.
- Sign Up:** A link at the bottom left of the form area.

Figure 66 Login

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Useranme	Username to login	No	Yes	TextBox	String
2	Password	Password to login	No	Yes	TexBox	String

Table 82 <Fields> Login

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
3	Login	Login into the system	Required	Transfer to Dashboard

Table 83 <Buttons/Hyperlinks> Login

5.1.2 Menu Manager

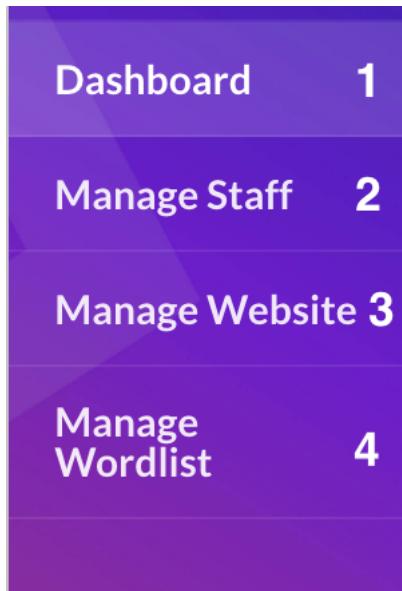


Figure 67 Menu Manager

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Dashboard	Dashboard	User must have been authorized.	Redirect to dashboard
2	Manage Staff	Manage staff	User must have one of these roles: Manager	Redirect to manage staff page
3	Manage Website	Manage website	User must have one of these roles: Manager	Redirect to manage website page
4	Manage Wordlist	Manage wordlist	User must have one of these roles: Manager	Redirect to manage word list page

Table 84 <Buttons/Hyperlinks> Menu Manage

5.1.3 Manage Staff

Manage Staff			
No	Information	Assign for Website	Action
1	Name: truong Username: staff	https://thanhnien.vn https://vnexpress.net	Edit Delete
2	Name: vi Username: staff2	https://thanhnien.vn	Edit Delete

Figure 68 Manage Staff

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
2	Search	Search field	No	Yes	TextBox	String
3	No	Staff's id	Yes	Yes	Text	String
4	Information	Staff's information	Yes	Yes	Text	String
5	Assign for website	Staff assigned website	Yes	Yes	Text	String

Table 85 <Fields> Manage Staff

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Add	Add new staff	N/A	Navigate to create staff page
7	Edit	Edit staff	N/A	Navigate to edit staff page
8	Delete	Delete staff	N/A	Delete staff Reload table

Table 86 <Buttons/Hyperlinks> Manage Staff

5.1.4 Manage Website

Manage Website						
1 Add						
ID 2	Name 3	URL 4	Version 5	Lastest updated 6	New version 7	Action
1	thanhnien	https://thanhnien.vn	3	7/11/2018	8 Make	9 Assign Edit 10 Delete 11
2	vnexpress	https://vnexpress.net	1	7/11/2018	Make	Assign Edit Delete

Figure 69 Manage Website

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
2	Id	Website id	Yes	Yes	Text	String
3	Name	Website name	Yes	Yes	Text	String
4	URL	Website URL	Yes	Yes	Text	String
5	Version	Website sitemap version	Yes	Yes	Text	String
6	Last updated	Time of the last sitemap updated	Yes	Yes	Text	String
7	New Version					

Table 87 <Fields> Manage Website

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Add	Add new website	N/A	Navigate to add website page.
8	Make	Make new website's sitemap	N/A	Reload table
9	Assign	Assign website for staff	N/A	Navigate to assign website for staff page

10	Edit	Edit website information	N/A	Navigate to edit website page
11	Delete	Delete website	N/A	Delete website Reload table

Table 88 <Buttons/Hyperlinks> Manage Website

5.1.5 Menu Staff

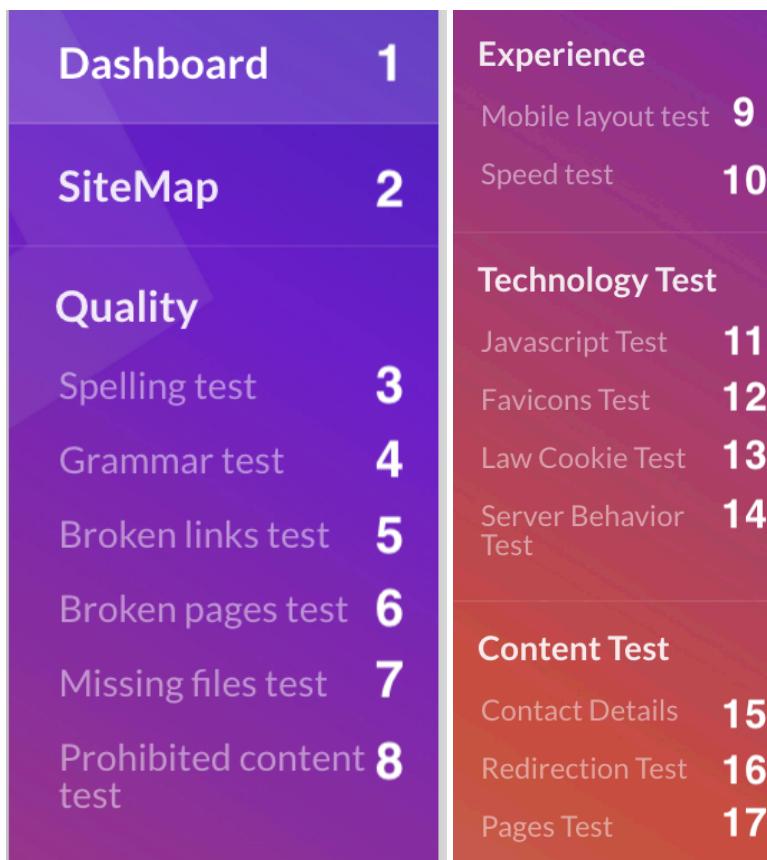


Figure 70 Menu Staff

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Dashboard	Dashboard	User must have been authorized.	Redirect to dashboard
2	Sitemap	Website sitemap	User must have one of these roles: staff	Redirect to sitemap page

3	Spelling test	Spelling test	User must have one of these roles: staff	Redirect to spelling test page
4	Grammar test	Grammar test	User must have one of these roles: staff	Redirect to grammar test page
5	Broken page test	Broken page test	User must have one of these roles: staff	Redirect to broken page test page
6	Broken link test	Broken link test	User must have one of these roles: staff	Redirect to broken link test page
7	Missing file test	Missing file test	User must have one of these roles: staff	Redirect to missing file test page
8	Prohibited content test	Prohibited content test	User must have one of these roles: staff	Redirect to prohibited content test page
9	Mobile Layout test	Mobile layout test	User must have one of these roles: staff	Redirect to mobile layout test page
10	Speed test	Measure the speed of each page	User must have one of these roles: staff	Redirect to Measure the speed of each page page
11	Javascript test	Javascript error test	User must have one of these roles: staff	Redirect to javascript error test page
12	Favicon test	Favicon test	User must have one of these roles: staff	Redirect to favion test page
13	Cookie test	Cookie test	User must have one of these roles: staff	Redirect to cookie test page
14	Server behavior test	Server behavior test	User must have one of these roles: staff	Redirect to server behavior test page

15	Contact details	Contact details test	User must have one of these roles: staff	Redirect to contact details test page
16	Redirection test	Redireaction test	User must have one of these roles: staff	Redirect to redirection test page
17	Page Test	Page test	User must have one of these roles: staff	Redirect to page test page

Table 89 <Buttons/Hyperlinks> Menu Staff

5.1.6 Sitemap

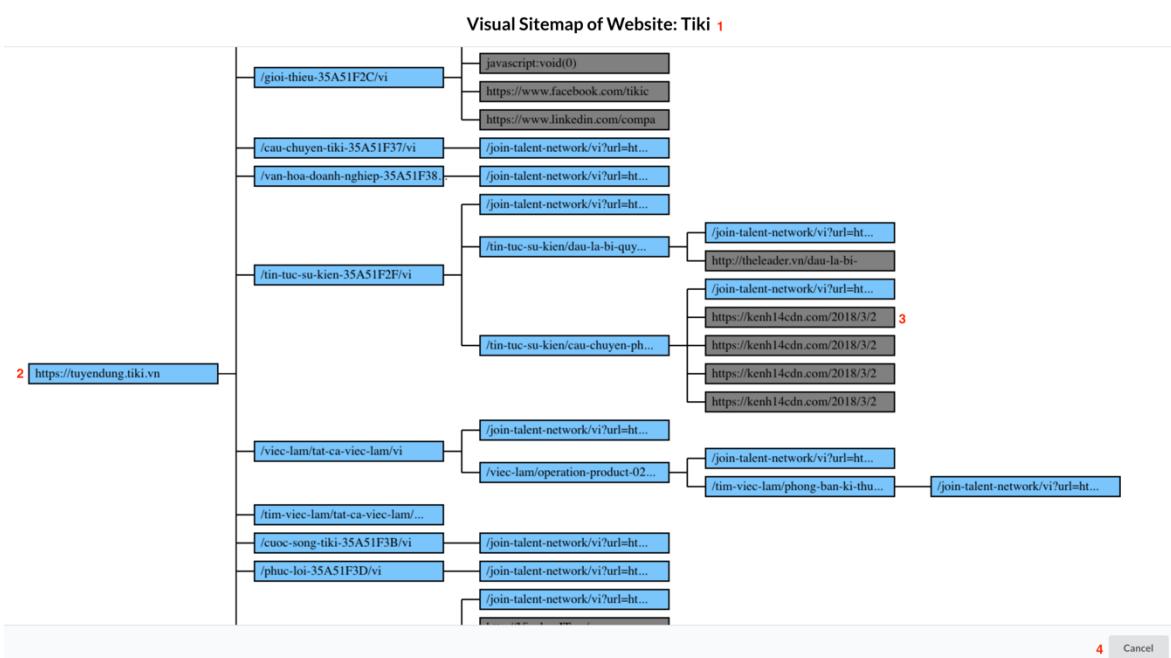


Figure 71 Sitemap

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Name Sitemap	Name sitemap	Yes	Yes	Text	String
2	Internal link	Internal link	Yes	Yes	Text	String
3	External link	External link	Yes	Yes	Text	String

Table 90 <Fields> Sitemap

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
4	Cancel	Close sitemap	N/A	Navigate to manage website.

Table 91 <Buttons/Hyperlinks> Sitemap

5.1.7 Spelling test

The screenshot shows a 'Spelling test' interface. At the top, there's a header with a logo of a cartoon notepad character holding a pencil. Below it, a toolbar has a 'Page option' button and a green box labeled '[Test Khanh]: 7 pages'. The main area displays a summary: 4 Unrecognized spellings and 2 Affected Pages. A table below lists four words with their suggestions and page counts, each with a 'Learn Spelling' button. The columns are labeled: Word, Suggestions, Found in, Pages, and Action.

3 Word	4 Suggestions	5 Found in	6 Pages	Action
suberd	suburd superd subduer	Text	1	7 Learn Spelling
suberd	suburd superd subduer	Text	1	Learn Spelling
suberd	suburd superd subduer	Text	1	Learn Spelling
suberd	suburd superd subduer	Text	1	Learn Spelling

Figure 72 Spelling Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Word	Word	Yes	Yes	Text	String
4	Suggestion	Suggestion word	Yes	Yes	Text	String
5	Found in	Found in	Yes	Yes	Text	String
6	Pages	Pages	Yes	Yes	Text	String

Table 92 <Fields> Spelling Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome

1	Page Option	Page option	N/A	Navigate to page option page.
5	Export	Export excel	N/A	Export to excel
7	Learn Spelling	Learn spelling	N/A	Reload table result

Table 93 <Buttons/Hyperlinks> Grammar Test

5.1.8 Grammar test

The screenshot shows a 'Grammar test' interface. At the top, there's a navigation bar with 'Page option' and '[Test Khanh]: 7 pages'. On the right, there's a cartoon character icon. Below the bar, a summary section shows '4 Potential grammar error' and '2 Affected Pages'. It includes a search bar and two examples of errors with suggested corrections ('Did you mean "as follows"?'). At the bottom right, there are 'Export' and 'Learn Grammar' buttons.

Figure 73 Grammar Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Excerpt	Excerpt	Yes	Yes	Text	String
4	Issue	Issue	Yes	Yes	Text	String

Table 94 <Fields> Grammar Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
5	Export	Export excel	N/A	Export to excel

7	Learn Grammar	Learn this grammar	N/A	Reload table result
---	---------------	--------------------	-----	---------------------

Table 95 <Buttons/Hyperlinks> Grammar Test

5.1.9 Broken page test

Broken pages test

This for test alt title



Page option 3 1

Check → 2

	0 Missing pages 3		0 Error pages 4
--	----------------------	--	--------------------

Search... Export 5

Page 6	Status 7	Issue 8
--------	----------	---------

This page haven't test yet, please try to test!

Figure 74 Broken Pages Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Missing pages	Missing pages	Yes	Yes	Text	String
4	Error pages	Error pages	Yes	Yes	Text	String
6	Pages	Page's url	Yes	Yes	Text	String
7	Status	http code response	Yes	Yes	Text	String
8	Issues	Broken page's issues	Yes	Yes	Text	String

Table 96 <Fields> Broken Pages Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
----	----------	-------------	------------	---------

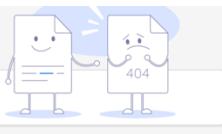
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
5	Export	Export excel	N/A	Export to excel

Table 97 <Buttons/Hyperlinks> Broken Pages Test

5.1.10 Broken links test

Broken links test

This test alt title



Page option 3 1

Check → 2

	0 Internal broken links 3		0 External broken links 4
--	------------------------------	--	------------------------------

Search... Export 5

Page 6	Broken link 7	Issue 8
This page has no broken links!		

Figure 75 Broken Links Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Internal Broken Links	Internal Broken Links	Yes	Yes	Text	String
4	External Broken Links	External Broken Links	Yes	Yes	Text	String
6	Pages	Page's url	Yes	Yes	Text	String
7	Broken link	Broken link	Yes	Yes	Text	String
8	Issues	Broken link's issues	Yes	Yes	Text	String

Table 98 <Fields> Broken Links Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
5	Export	Export excel	N/A	Export to excel

Table 99 <Buttons/Hyperlinks> Broken Links Test

5.1.11 Missing file test

Figure 76 Missing File Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Number of missing files	Numbers of missing file	Yes	Yes	Text	String
4	Number of affected pages	Number of affected pages	Yes	Yes	Text	String
9	Search	Search field	No	Yes	TextBox	String
11	Files	File missing	Yes	Yes	Text	String

12	Description	Description missing	Yes	Yes	Text	String
13	Pages	Pages affected	Yes	Yes	Text	String
14	Action	Edit	Yes	Yes	Text	String

Table 100 <Fields> Missing File Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
5	Image	Image option	N/A	Button image is active
6	CSS	CSS option	N/A	Button css is active
7	DOC	DOC option	N/A	Button doc is active
8	Archires	Archires option	N/A	Button arrhires is active
10	Export	Export excel	N/A	Export to excel

Table 101 <Buttons/Hyperlinks> Missing File Test

5.1.12 Prohibited content test

The screenshot shows a web-based application for prohibited content testing. At the top, there's a header "Prohibited content test" and a note "This for test alt title". Below the header is a search bar with a magnifying glass icon and a red "X" button. To the right of the search bar is a "Search..." input field and an "Export" button with a file icon and the number "4". On the left, there's a "Page option" button with a counter "0" and a red "1" indicating new items. Below the search bar, there are four tabs labeled "Word 5", "Type 6", "Fragment 7", and "Page 8". Under each tab, there's a table with three rows. The first row contains "gun" under Word, "Weapons" under Type, "gun case with 5 mag pockets 4." under Fragment, and a link "https://www.sportsmansguide.com" under Page. The second row contains "rifle" under Word, "Weapons" under Type, "rifle case 5 stars 5 (1) buyer" under Fragment, and a link "https://www.sportsmansguide.com" under Page. The third row contains "rifle" under Word, "Weapons" under Type, "rifle scope 5 stars 5 (1) buye" under Fragment, and a link "https://www.sportsmansguide.com" under Page. A large red "X" icon is overlaid on the top right of the interface.

Figure 77 Prohibited content test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
5	Word	Prohibited word	Yes	Yes	Text	String
6	Type	Type of prohibited word	Yes	Yes	Text	String
7	Fragment	Fragment includes prohibited word	Yes	Yes	Text	String
8	Page	Page's url	Yes	Yes	Text	String

Table 102 <Fields> Prohibited Content Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
3	Save	Save	N/A	Reload table
4	Export	Export excel	N/A	Export to Excel

Table 103 <Buttons/Hyperlinks> Prohibited Content Test

5.1.13 Mobile layout test

Mobile layout test

This test alt title

Page option 1

Check → 2

Search... 3 Export

ScreenShot 4

Page 5

Issue 6

Digital Marketing - Marketing Online|Tin Típ Thị Sô 24h | GenK.vn
http://genk.vn/digital-marketing.chn

Not Optimize for Mobile!

Figure 78 Mobile Layout Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
4	Screenshoot	Screenshoot website mobile layout	Yes	Yes	Text	String
5	Pages	Page's url	Yes	Yes	Text	String
6	Issues	Mobile layout's issue	Yes	Yes	Text	String

Table 104 <Fields> Mobile Layout Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
3	Export	Export excel	N/A	Export to excel

Table 105 <Buttons/Hyperlinks> Mobile Layout Test

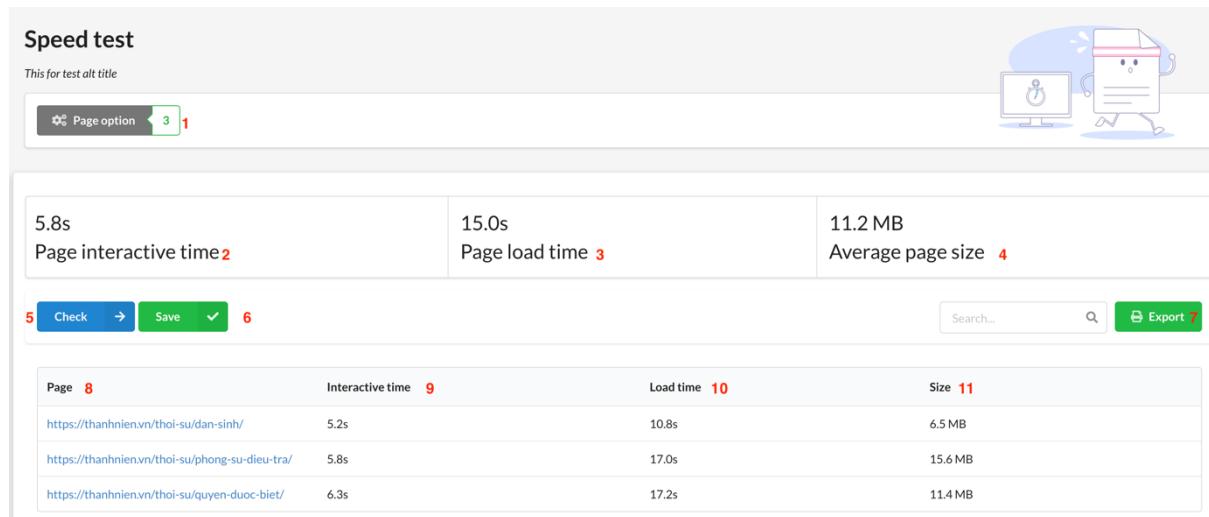
5.1.14 Speed test

Figure 79 Speed Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
2	Page Interactive time	Average page interactive time	Yes	Yes	Text	String
3	Page load time	Average page load time	Yes	Yes	Text	String
4	Average page size	Average page size	Yes	Yes	Text	String
8	Page	Page's url	Yes	Yes	Text	String
9	Interactive time	Page interactive time	Yes	Yes	Text	String
10	Load time	Page load time	Yes	Yes	Text	String
11	Size	Page size	Yes	Yes	Text	String

Table 106 <Fields> Speed Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
5	Check	Check	N/A	Reload table
6	Save	Save	N/A	Reload table
7	Export	Export excel	N/A	Export to excel

Table 107 <Buttons/Hyperlinks> Speed Test

5.1.15 Javascript error test

The screenshot shows a web-based application for testing website errors. At the top, there's a header with the title "Javascript Test". Below it, a sub-header says "This for test alt title". A toolbar includes "Page option" (with a dropdown showing "3" and "1"), "Check" (with a dropdown showing "2"), "Search..." (with a magnifying glass icon), and "Export" (with a file icon). The main content area displays two error messages. Each message has a "Type" field (set to "5") and a "Pages" field (set to "6"). The first message is a "WARNING" about a parserblocking cross-site script from securepubads.g.doubleclick.net. It includes a link to https://www.chromestatus.com/feature/5718547946799104. The second message is another "WARNING" about a similar script, also with a link to https://www.chromestatus.com/feature/5718547946799104.

Figure 80 - JavaScript Errors Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
4	Error Message	Javascript Error's information	Yes	Yes	Text	String
5	Type	Error type	Yes	Yes	Text	String
6	Pages	Page's url	Yes	Yes	Text	String

Table 108 <Fields> JavaScript Errors Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
3	Export	Export excel	N/A	Export to excel

Table 109 <Buttons/Hyperlinks> JavaScript Errors Test

5.1.16 Favicon test

The screenshot shows the 'Favicons Test' interface. At the top, there's a header with the title 'Favicons Test' and a note 'This for test alt title'. Below it is a toolbar with a 'Page option' button, a green '0' button, and a red '1' button. To the right is a status bar with icons for 'Slikide', 'Secure', and the URL 'https://slikide.com'.

The main area has a 'Check' button with a red '2' and a green checkmark icon. Below this are two rows of results:

- Row 1:** A purple star icon, '1 Different Favicons 3', and a green checkmark icon with '1 Pages missing Favicons 4'.
- Row 2:** A search bar with 'All Favicons', a search icon, and an 'Export' button with a red '6'.

At the bottom, there's a detailed table row with columns for 'Favicon 7', 'URL 8', 'Width 9', and 'Pages 10'. It shows an image of a blue circle with 'IN', the URL 'https://insites.com/app/themes/insites/apple-icon-57x57.png', a width of '96x96', and the page URL 'https://insites.com/content/redirections/'.

Figure 81 Favicon Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Number of different favicons	Numbers of different favicons	Yes	Yes	Text	String
4	Number of pages missing favicons	Number of pages missing favicon	Yes	Yes	Text	String
5	Search	Search field	No	Yes	TextBox	String
7	Favicon	Favicon image	Yes	Yes	Text	String
12	URL	Favicon URL	Yes	Yes	Text	String
13	Width	Favicon width	Yes	Yes	Text	String
14	Pages	Pages affected	Yes	Yes	Text	String

Table 110 <Fields> Favicon Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome

1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
6	Export	Export excel	N/A	Export to excel

Table 111 <Buttons/Hyperlinks> Favicon Test

5.1.17 Cookie test

The screenshot shows a web application titled "Law Cookie Test". At the top left, there is a note: "This for test alt title". Below the title, there is a navigation bar with a "Page option" button and a "Check" button. A green box highlights the number "1" next to the "Page option" button. To the right of the navigation bar is a search bar and an "Export" button with the number "3" next to it. The main content area displays a table of cookie data. The columns are labeled: "Cookie Name" (highlighted with number "4"), "Category" (highlighted with number "5"), "Party" (highlighted with number "6"), and "Description" (highlighted with number "7"). The table contains two rows of data. The first row has values ".ASPXANONYMOUS", "Unknown", ".thanhnien.vn", and "The purpose of these cookies is unknown.". The second row has values "_R", "Unknown", "thanhnien.vn", and "The purpose of these cookies is unknown.". The entire screenshot is framed by a light gray border.

Figure 82 Cookie Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
4	Cookie name	Cookie name	Yes	Yes	Text	String
5	Category	Cookie category	Yes	Yes	Text	String
6	Party	Cookie party	Yes	Yes	Text	String
7	Description	Cookie description	Yes	Yes	Text	String

Table 112 <Fields> Cookie Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table

3	Export	Export excel	N/A	Export to excel
---	--------	--------------	-----	-----------------

Table 113 <Buttons/Hyperlinks> Cookie Test

5.1.18 Server behavior test

Server Behavior Test 

1  [Test Khanh]: 7 pages

Done 0

Website redirects from www to non-www.
The website should handle URLs with a www prefix and redirect them to a non-www prefix wherever possible. 2

Website redirects HTTP to HTTPS.
The website should redirect non-SSL pages to SSL pages wherever possible. 3

All pages use SSL.
Google is progressively penalising websites which avoid SSL, in both search results and the Chrome browser. 4

Figure 83 Server Behavior

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
2	Website redirects from www to non-www	Website redirects from www to non-www	Yes	Yes	Text	String
3	Website redirects from HTTP to HTTPS	Website redirects from HTTP to HTTPS	Yes	Yes	Text	String
4	All page use SSL	All page use SSL	Yes	Yes	Text	String

Table 114 <Fields> Server Behavior

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.

Table 115 <Buttons/Hyperlinks> Server Behavior

5.1.19 Contact detail test

The screenshot shows a web-based application titled "Contact Details". At the top, there is a header with the title and some status indicators. Below the header is a toolbar with icons for "Page option" (with a count of 0), "Check" (with a count of 2), and "Export". The main area contains a table with two rows. The first row has icons for a phone and an envelope, followed by the number "19" and the text "Phone Numbers 3". The second row has an icon for an envelope, followed by the number "5" and the text "Emails 4". At the bottom of the table, there is a footer with the text "Contact 6", "Page affected 7", and "Action". The "Action" column contains a link to "https://www.orderhiccupsteahouse.com/" and an "Edit 8" button.

Figure 84 - Contact Details Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Number of phone numbers	Numbers of phone numbers	Yes	Yes	Text	String
4	Number of emails	Number of emails	Yes	Yes	Text	String
6	Contact	Email or phone numbers	Yes	Yes	Text	String
7	Page affected	Page's URL affected	Yes	Yes	Text	String

Table 116 <Fields> Contact Details

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
5	Export	Export excel	N/A	Export to excel

8	Edit	Edit	N/A	Reload table
---	------	------	-----	--------------

Table 117 <Buttons/Hyperlinks> Contact Details Test

5.1.20 Redirection test

Redirection Test

This for test alt title

Page option 0 1

Check → 2

Web Address 5

3 Export Search... 4

6 Directs to 7 Type 8 Code

This page haven't test yet, please try to test

Figure 85 - Redirection Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
4	Search	Search fields	No	Yes	TextBox	String
5	Web address	Website address	Yes	Yes	Text	String
6	Directs to	Direct to URL	Yes	Yes	Text	String
7	Type	Type redirection	Yes	Yes	Text	String
8	Code	HTTP code response	Yes	Yes	Text	String

Table 118 <Fields> Redirection Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
3	Export	Export excel	N/A	Export to excel

Table 119 <Buttons/Hyperlinks> Redirection Test

5.1.21 Page test

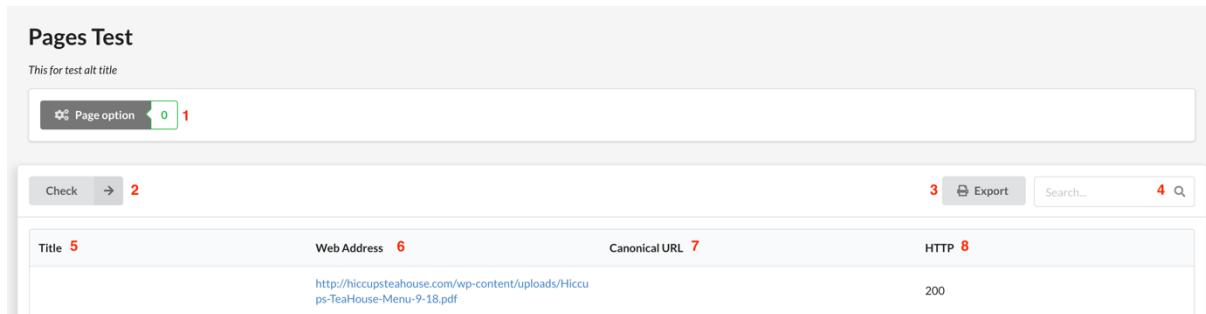


Figure 86 Page Test

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
4	Search	Search fields	No	Yes	TextBox	String
5	Title	Website title	Yes	Yes	Text	String
6	Web address	Website address	Yes	Yes	Text	String
7	Canonical URL	Canonical URL	Yes	Yes	Text	String
8	HTTP	HTTP code response	Yes	Yes	Text	String

Table 120 <Fields> Page Test

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Page Option	Page option	N/A	Navigate to page option page.
2	Check	Check	N/A	Reload table
3	Export	Export excel	N/A	Export to excel

Table 121 <Buttons/Hyperlinks> Page Test

6. Database Design

6.1. Entity relationship diagram (ERD)

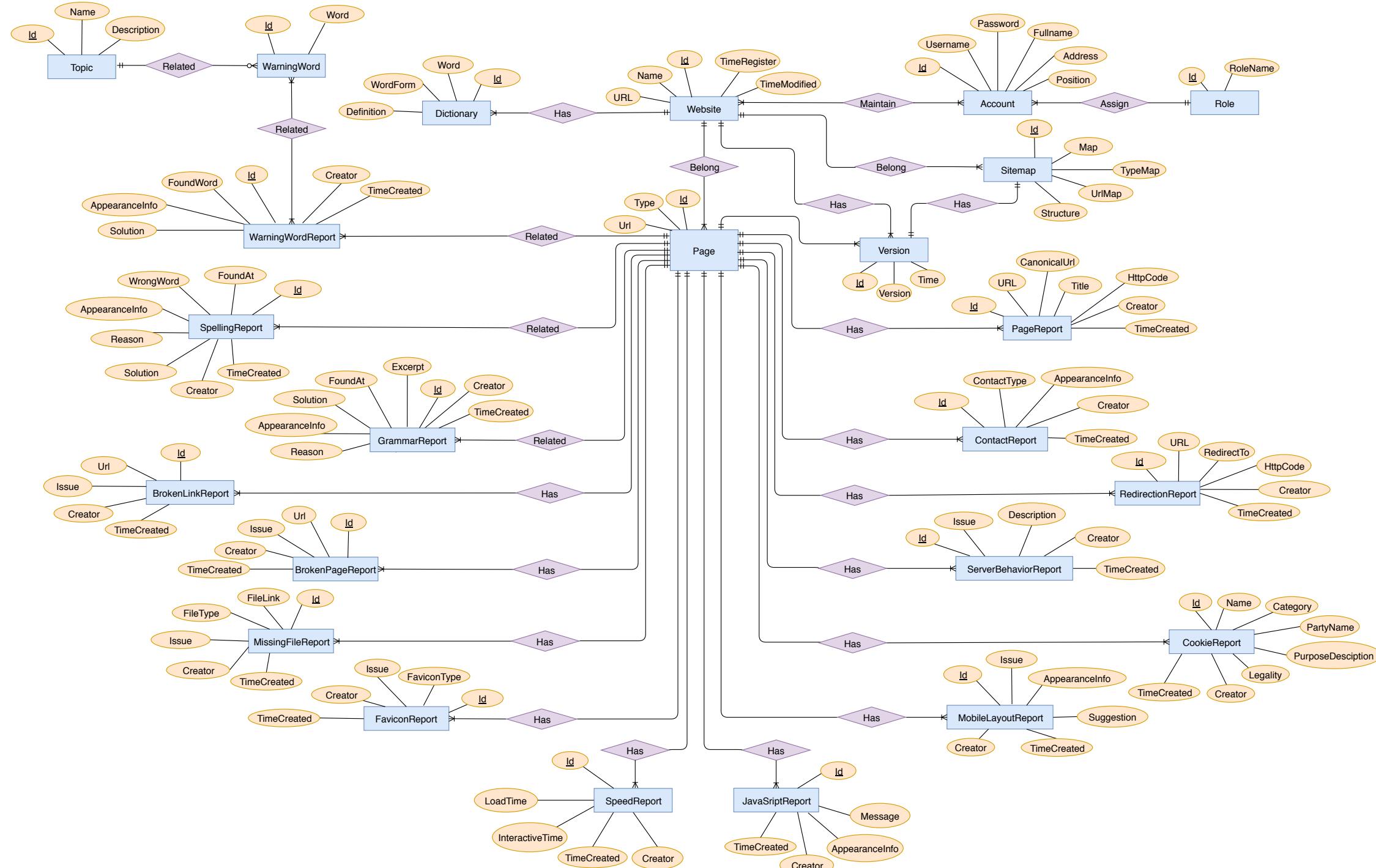


Figure 87 Entity Relationship Diagram

6.2. Data Dictionary

Entity Data Dictionary: describe content of all entities	
<u>Entity name</u>	<u>Description</u>
Account	Contains the account's information.
Role	Contains the role's information.
Sitemap	Contains the sitemap's information.
Version	Show the version of Sitemap.
Page	Contains the page's information.
Website	Contains the website's information.
Dictionary	Contains the dictionary's information.
Topic	Contains the topic's information.
WarningWord	Contains the warningWord's information
WarningWordReport	Contains the warningWordReport's information.
SpellingReport	Contains the spellingReport information
GrammarReport	Contains the grammarReport's information.
BrokenLinkReport	Contains the brokenLinkReport's information.
BrokenPageReport	Contains the brokenPageReport's information.
MissingFileReport	Contains the missingFileReport information.
FaviconReport	Contains the faviconReport's information.
SpeedReport	Contains the speedReport's information.

JavaScriptReport	Contains the javaScriptReport's information.
MobileLayoutReport	Contains the mobileLayoutReport's information.
CookieReport	Contains the cookieReport's information.
ServerBehaviorReport	Contains the serverBehaviorReport's information.
RedirectionReport	Contains the redirectionReport's information.
ContactReport	Contains the contactReport's information.
PageReport	Contains the pageReport's information.

Table 122 Entity Relationship Diagram Dictionary

Entity Name	Attributes	Description	Domain	Null
account	id	Id of account	BIGINT(20)	NO
	position	Position of account	VARCHAR(255)	NO
	address	Address of account	VARCHAR(255)	NO
	fullname	Fullscreen of account	VARCHAR(255)	NO
	password	Password of account	VARCHAR(255)	NO
	username	Username of account	VARCHAR(255)	NO
role	id	Role id of account	BIGINT(20)	NO
	role_name	Role name of account	VARCHAR(255)	NO
sitemap	id	Id of site map	BIGINT(20)	NO
	map	Map of site map	VARCHAR(255)	NO
	type_map	Type of site map	VARCHAR(255)	NO
	url_map	Url of site map	VARCHAR(255)	NO

version	id	Id of version	BIGINT(20)	NO
	version	Number of version.	BIGINT(20)	NO
	time	Time created version	DATETIME	NO
page	id	Id of page	BIGINT(20)	NO
	type	Type of page	BIGINT(20)	NO
	url	Url of page	VARCHAR(255)	NO
website	id	Id of website	BIGINT(20)	NO
	name	Name of website	VARCHAR(255)	NO
	url	Url of website	VARCHAR(255)	NO
	time_register	Time register of website	DATETIME	NO
	time_modified	Time Modified of website	DATETIME	NO
topic	id	Id of topic	BIGINT(20)	NO
	name	Name of topic	VARCHAR(255)	NO
	description	Description of topic	VARCHAR(255)	NO
warningWord	id	Id of warning word	BIGINT(20)	NO
	word	Warning word	VARCHAR(255)	NO
dictionary	id	Id of dictionary	BIGINT(20)	NO
	word	Word in dictionary	VARCHAR(255)	NO
	word_form	Word form of dictionary	VARCHAR(255)	NO
	definition	Definition of word	VARCHAR(255)	NO
warningWordReport	id	Id of warning word report	BIGINT(20)	NO
	found_word	Word found	VARCHAR(255)	NO
	appearance_info	Appearance info	VARCHAR(255)	NO
	solution	Solution	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO

	time_created	Time to create	DATETIME	NO
spellingReport	id	Id of spelling report	BIGINT(20)	NO
	found_at	Position found	VARCHAR(255)	NO
	wrong_word	Wrong word	VARCHAR(255)	NO
	appearance_info	Appearance info	VARCHAR(255)	NO
	reason	Reason wrong	VARCHAR(255)	NO
	solution	Solution	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
grammarReport	id	Id of grammar report	BIGINT(20)	NO
	excerpt	Excerpt of grammar report	VARCHAR(255)	NO
	found_at	Where found error	VARCHAR(255)	NO
	appearanceInfo	Appearance info	VARCHAR(255)	NO
	reason	Reason wrong	VARCHAR(255)	NO
	solution	Solution	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
brokenLinkReport	id	Id of broken link report	BIGINT(20)	NO
	url	url of broken link	VARCHAR(255)	NO
	issue	Issue of broken link	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO

brokenPageReport	id	Id of broken page report	BIGINT(20)	NO
	url	issue of broken page	VARCHAR(255)	NO
	issue	Issue of broken page	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
missingFileReport	id	Id of missing file report	BIGINT(20)	NO
	file_link	Link file error	VARCHAR(255)	NO
	file_type	Type file error	VARCHAR(255)	NO
	issue	Issue of file error	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
faviconReport	id	Id of favicon Report	BIGINT(20)	NO
	favicon_type	Type of favicon	VARCHAR(255)	NO
	issue	Issue of favicon	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
speedReport	id	Id of speed report	BIGINT(20)	NO
	load_time	Time to load	VARCHAR(255)	NO
	interactive_time	Time to interactive	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
JavaScriptReport	id	Id of JavaScript report	BIGINT(20)	NO
	message	Message error	VARCHAR(255)	NO
	appearance_info	Appearance info	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
mobileLayoutReport	id	Id of mobile layout report	BIGINT(20)	NO
	issue	Issue of mobile layout	VARCHAR(255)	NO

	appearance_info	Appearance info	VARCHAR(255)	NO
	suggestion	Suggestion for user	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
cookieReport	id	Id of cookie report	BIGINT(20)	NO
	name	Message content	VARCHAR(255)	NO
	category	Category of cookie	VARCHAR(255)	NO
	party_name	Party name of cookie	VARCHAR(255)	NO
	purpose_descript ion	Purpose of cookie	VARCHAR(255)	NO
	legality	Legality of file	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
serverBehaviorReport	Id	Id of server behavior report	BIGINT(20)	NO
	issue	Issue of server	VARCHAR(255)	NO
	description	Description server behavior	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
redirectionReport	id	Id of redirection Report	BIGINT(20)	NO
	url	url of link	VARCHAR(255)	NO
	redirect_to	url redirect to	VARCHAR(255)	NO
	http_code	Http code of link	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
contactReport	id	Id of contact report	BIGINT(20)	NO

	contact_type	Type of contact	VARCHAR(255)	NO
	appearance_info	Appearance info	VARCHAR(255)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO
pageReport	id	Id of page report	BIGINT(20)	NO
	url	url of page	VARCHAR(255)	NO
	canonical_url	Canonical url of page	VARCHAR(255)	NO
	title	Title of page	VARCHAR(255)	NO
	httpCode	Http code of page	BIGINT(20)	NO
	creator	Who create	DATETIME	NO
	time_created	Time to create	DATETIME	NO

7. Algorithms

7.1. Sitemap Generator

a. Definition:

Enqueue each level of a tree sequentially as the root of any subtree is encountered. There are 2 cases in the iterative algorithm:

- **Root case:** The traversal queue is initially empty so the root node must be added before the general case.
- **General case:** Process any items in the queue, while also expanding their children. Stop if the queue is empty. The general case will halt after processing the bottom level as leaf nodes have no children.

Input: the root URL of a website

Output: The sitemap.

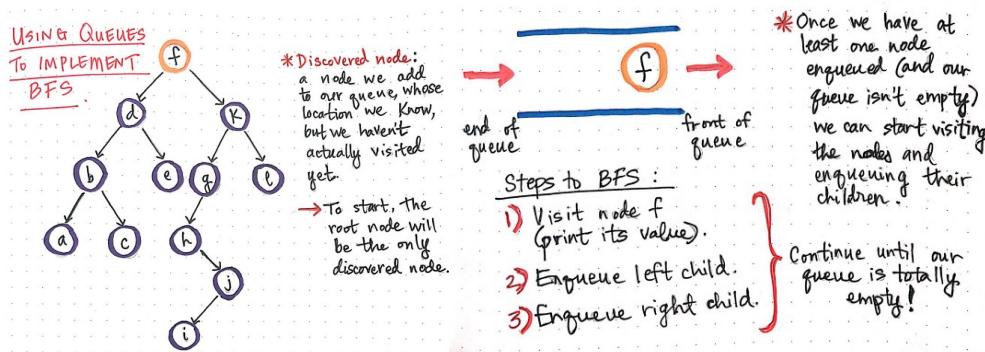


Figure 88 Define Sitemap

❖ Implementation

```

queue.add (rootURL);
urlMap.put(root, vertice_number)
vertice_number++;
graph.add(newArrayList);
typeList.add(1);
linkList.add(root);

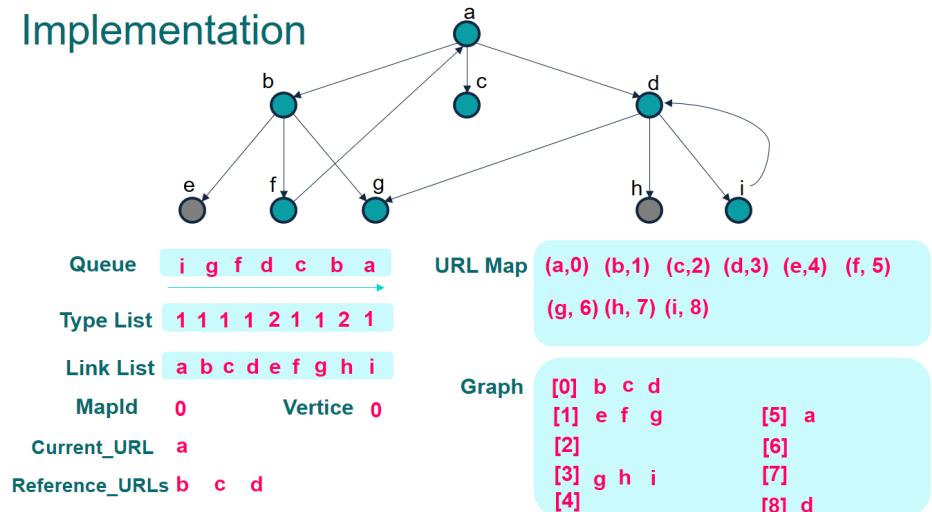
```

```

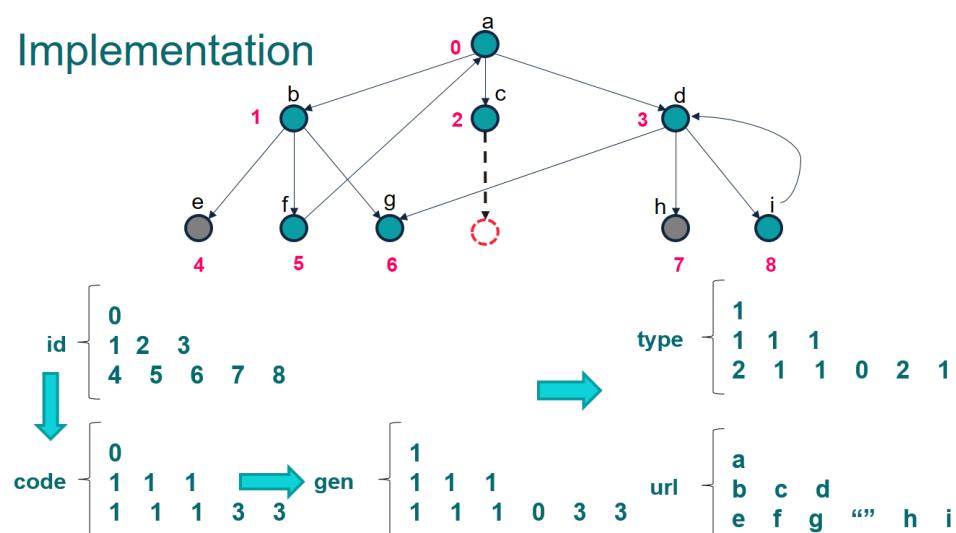
while queue is not empty
    current_url = queue.poll()
    mapId = urlMap.get(current_url)
    refURLs = getListReferences(current_url )
    for url in refURLs:
        if urlMap.notContainKey(url)
            urlMap.put(url, vertice_number);
            vertice_number++;
            graph.add(new ArrayList);
            if url is internal link
                site = new SiteLink(source_link: current_url, des_link: url, type: 1);
                graph.get(mapId).add(site);
                typeList.add(1);
                linkList.add(url);
                queue.add(url);
            else (url is external link)
                site = new SiteLink(source_link: current_url, des_link: url, type: 2);
                graph.get(mapId).add(site);
                typeList.add(2);
                linkList.add(url);
            else (existed node)
                current_map_index = urlMap.get(url);
                current_type_url = typeList.add(current_map_index);
                site = new SiteLink(source_link: current_url, des_link: url, current_type_url);
                graph.get(mapId).add(site);

```

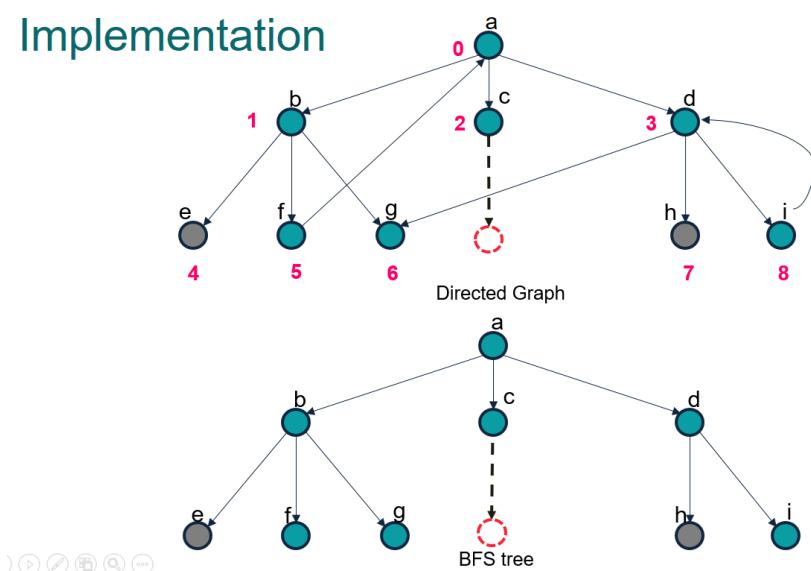
Implementation



Implementation

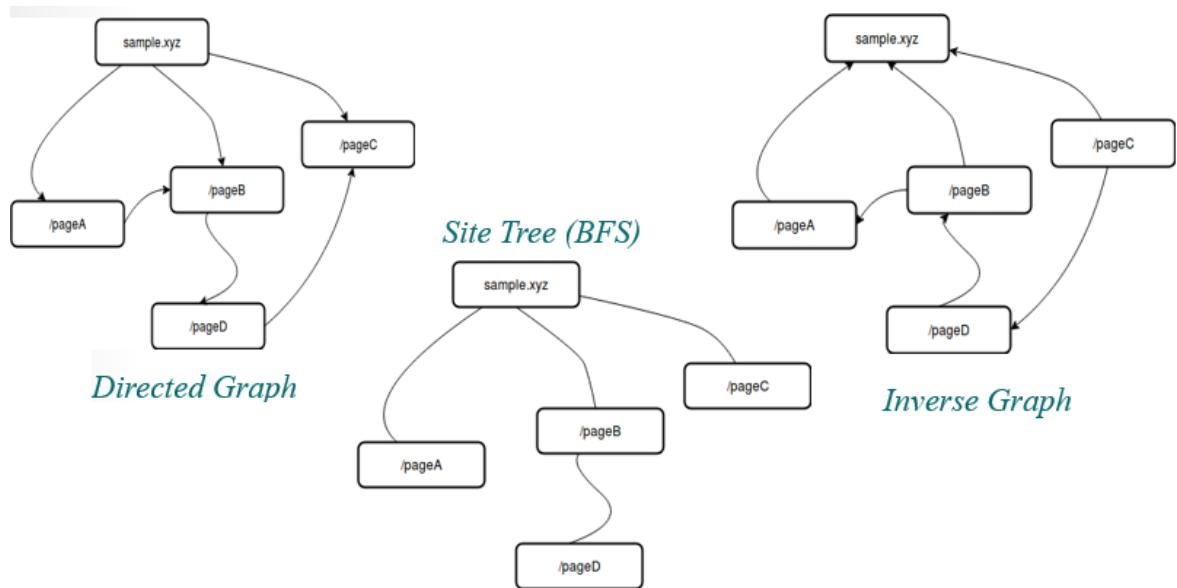


Implementation



b. Sitemap Structures

- SiteLink: object with attributes:
 - src: source link
 - des: destination link
 - desType: type of destination link (internal, external)
- HashMap<String, Integer> urlMap: Number URLs into distinct integers, which is also the vertex index on the graph.
- List<List<SiteLink>> graph: adjacency list
- List<List<SiteLink>> invGraph: adjacency list represents graph with all inverse edges from origin graph.
- List<Integer> urlType: get type of link from vertice index.
- List<String> links: inverse mapping with urlMap, allow get url link from vertice index.
- codeGraph: string encoded of graph to create visual image.



Define some functions

- CreateEdge(u, v): create edge from u to v and add to graph, also create an inverse edge v to u and store to invGraph.
- getListReferences(url): crawl with url and list all others url links in that page. We could also get more information like urlType.
- getUniqueId(): get new unique integer number.

Pseudo code for BFS

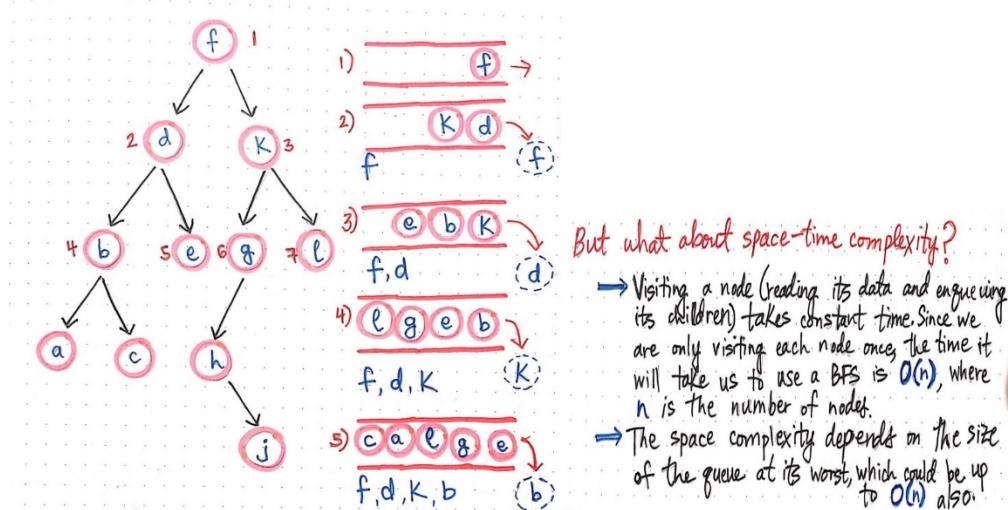
```

1: function BFS(rootURL)
2:   Queue q → init empty
3:   urlMap.mark(rootURL) = getUniqueID()
4:   while q is not empty
5:     URL = q.pool()
6:     refURLs = getListReferences(URL)
7:     for url in refURLs:
8:       createEdge(URL, urlMap) // create edge of graph
9:       if urlMap.notContainKey(url)
10:        urlMap.mark(url) = getUniqueID()
11:        q.add (url)

```

Note:

- The depth of the sitemap-tree is always minimal
- A tree path from one vertex to another is a way to get to the desired link through the link.



7.2. Spelling test

a. Definition Trie Tree

A **trie** is a tree-like data structure whose nodes store the letters of an alphabet. By structuring the nodes in a particular way, words and strings can be retrieved from the structure by traversing down a branch path of the tree.

We know that tries are often used to represent words in an alphabet. In the illustration shown here, we can start to get a sense of how exactly that representation works.

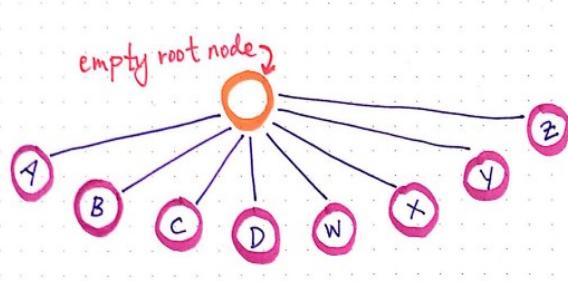


Figure 89 Trie Tree

Each trie has an empty root node, with links (or references) to other nodes—one for each possible alphabetic value.

The shape and the structure of a trie is always a set of linked nodes, connecting back to an empty root node. An important thing to note is that the number of child nodes in a trie depends completely upon the total number of values possible. For example, if we are representing the English alphabet, then the total number of child nodes is directly connected to the total number of letters possible. In the English alphabet, there are 26 letters, so the total number of child nodes will be 26.

b. Structures of Trie node

- `isLeaf`: determine true if the path from root to current node could build a correct word in dictionary.
- `eWeight`: the frequently of failure when user type wrong.
- `refNode`: List of reference link to another Trie Node, normally, Trie Node has 26 reference links corresponding to 'a' --> 'z', we can extend the number of reference link if the collection of character has larger.

A single node in a trie contains just two things:

1. A value, which might be null
2. An array of references to child nodes, all of which also might be null

Each node in a trie, including the root node itself, has only these two aspects to it. When a trie representing the English language is created, it consists of a single root node, whose value is usually set to an empty string: "".

That root node will also have an array that contains 26 references, all of which will point to null at first. As the trie grows, those pointers start to get filled up with references to other nodes nodes.

The way that those pointers or references are represented is particularly interesting. We know that each node contains an array of references/links to other nodes. What's cool about this is that we can use the array's indexes to find specific references to nodes. For example, our root node will hold an array of indexes 0 through 25, since there are 26 possible slots for the 26 letters of the alphabet. Since the alphabet is in order, we know that the reference to the node that will contain the letter A will live at index 0.

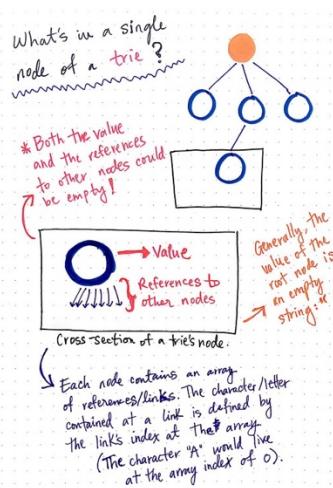


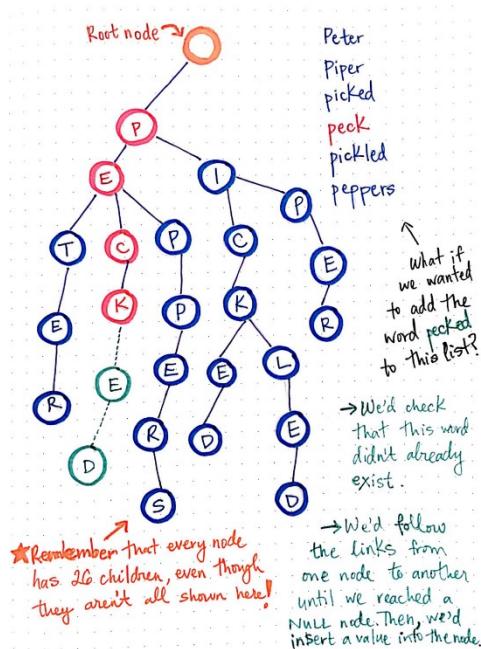
Figure 90 Structure Trie Tree

Looking at our trie, we can see that we have an empty root node, as is typical for a trie structure. We also have six different words that we're representing in this trie: Peter, piper, picked, peck, pickled, and peppers.

To make this trie easier to look at, I've only drawn the references that actually have nodes in them; it's important to remember that, even though they're not illustrated here, every single node has 26 references to possible child nodes.

Notice how there are six different “branches” to this trie, one for each word that's being represented. We can also see that some words are sharing parent nodes. For example, all of the branches for the words Peter, peck, and peppers share the nodes for p and for e. Similarly, the path to the word picked and pickled share the nodes p, i, c, and k.

So, what if we wanted to add the word pecked to this list of words represented by this trie? We'd need to do two things in order to make this happen:



First, we'd need to check that the word pecked doesn't already exist in this trie.

Next, if we've traversed down the branch where this word ought to live and the word doesn't exist yet, we'd insert a value into the node's reference where the word should go. In this case, we'd insert e and d at the correct references.

But how do we actually go about checking if the word exists? And how do we insert the letters into their correct places?

- **Idea of inserting method:** go through each character of word, check if has reference link correspond with this character. If not, we make new reference link. If the link has, just follow the path. Finally, mark the current node as a Leaf - that mean path from root to this node form correct word in dictionary.

Pseudo code for Insert word to Trie Tree

```

1: function mapPosition(char letter)
2:   return (letter - 'a')
3:
4: function insertWord(String word)
5:   Node currentNode = root
6:   for each char w of word:
7:     position = mapPosition(w)
8:     if currentNode.getRefNode(position) = null then

```

- ```

9: currentNode.setRefNode(position, new Node())
10: currentNode = currentNode.getRefNode(position)
11: currentNode.isLeaf = True

```
- **Idea id searching on Trie Tree:** The method is quite same the insert method but when we check in some Trie Node if doesn't contain reference link, just return with False result. If could find the path, check if this Trie Node is Leaf.

### **Pseudo code for Search word in Trie Tree**

```

1: function searchWord(String word)
2: Node currentNode = root
3: for each char w of word:
4: position = mapPosition(w)
5: if currentNode.getRefNode(position) = null then
6: return False
7: currentNode = currentNode.getRefNode(position)
8: return currentNode.isLeaf

```

### **c. Spelling generator using recursion**

We come up with the smallest case: wrong one letter.

That mean, if the needed check word has wrong one letter (missing, redundant, difference), the system will provide some suggestion words in dictionary.

Normal method is “brute force”, we can generate all cases of a word could be, by adding one letter to word: happ → happ + y, by changing one word: happ → harp, or remove one letter: happ → hap.

We make generator use recursion:

Let function **Generator** (word, position, currentGenWord, changed) is a state when we generate word with:

- word: the word we need to generate to all possible
- position: we stand at word.charAt(position), if position > word.length the method terminate.
- currentGenWord: current word that we are constructing, when postion reach the end of word, then currentGenWord is finished made.

- changed: the flag to verify that if currentGenWord is differ one letter with original word.

Transform state:

- if **changed** = False:
  - We couldn't make any change to origin **word**, so just add:
    - currentGenWord → currentGenWord + word.charAt(position)
    - position → position + 1
  - if **changed** = True:
    - We could do nothing one word, it's same case above when **changed** is False. The difference that **changed** still get True.
    - We could remove current letter by just increase **position** and go the another state:
      - position → position + 1
      - changed → False
    - We could add one letter to current **position** and make no change **position**:
      - currentGenWord → currentGenWord + (any char from 'a' to 'z')
      - changed → False
    - We could change current letter:
      - currentGenWord → currentGenWord + replaceOther(word.charAt(position))
      - changed → False

Finally, we get the generator for wrong one letter. We could improve the method to flexible extend by changing the flag **changed** from boolean value to int value with mean, the remain of change we could make on origin **word**. Then if we check wrong for two or three letters, just pass the param **changed** = 2 or 3 when call the recursion.

After have generator, we combine with any searching method, we could make a list of suggestion words. Currently, we compare the speed with two searches method:

- Linear Search
- Trie Tree Search

The Trie Tree Search's speed dominate of Linear Search, faster than about 150 times with 10 input words. Because the time to search one word in Trie Tree equal

length of input word when we have to iterator all words in dictionary if using Linear Search.

But, we still could improve method better by combine Trie Tree Search on recursion generator: we keep a Trie Node go follow with recursion. We can terminate if it current Trie Node is null.

The recursion method now look like: **Generator** (word, position, currentGenWord, changed, currentTrieNode).

#### Pseudo code for Generator combine Trie Tree search

1: **function** Generator(word, position, currentGenWord, changed, currentTrieNode)

2: **if** currentTrieNode = null **return** // terminate condition

3: **if** position > word.length **return** // terminate condition

4: **for** all **of** possible states:

**Generator** (word, new position, new currentGenWord, new changed, new TrieNode)

Note that: new TrieNode could get simple by follow the modify letter: if we add letter 'a', the new TrieNode will equal link from current Trie Node to refNode.get('a').

The method after optimizing is quite fast than normal search. We could see that the method will stop when could not found on tree that prevent to generate all cases that wasted a lot of time.

| Detection | Trie and Recursion | Trie Tree Search | HashSet Search | Normal Search                |
|-----------|--------------------|------------------|----------------|------------------------------|
| 1 st      | 171.665            | 228.58           | 360.905        | 70.843.211                   |
| 2 nd      | 282.936            | 515.356          | 486.516        | 213.282.707                  |
| 3 rd      | 227.485            | 424.986          | 356.754        | 217.496.979                  |
| 4 th      | 133.382            | 322.748          | 438.503        | 346.019.168                  |
| 5 th      | 105.295            | 131.974          | 225.477        | 251.235.083                  |
| 6 th      | 87.536             | 131.619          | 218.067        | 295.021.079                  |
| 7 th      | 122.989            | 151.977          | 213.425        | 245.818.985                  |
| 8 th      | 134.28             | 363.798          | 305.237        | 303.394.747                  |
| 9 th      | 102.281            | 236.726          | 197.911        | 252.620.838                  |
| 10 th     | 246.47             | 437.805          | 295.182        | 303.870.749                  |
| Finish    | 277.041.472        | 732.268.077      | 930.599.743    | 2.568.054.697.004 (nano-sec) |
|           | 0.277              | 0.732            | 0.93           | 2568 (seconds)               |
|           |                    |                  |                | 42.8 (minutes)               |

## E. System Implementation & Test

### 1. Introduction

#### 1.1. Overview

This section describes approach and methodologies used by our team to plan, organize and manage testing process of WCS.

It provides all necessary information about implementation and testing procedure of the system including test plans, test cases, test results and pass/fail criteria as well as testing flow to cover all possible cases.

#### 1.2. Test Approach

- **Goal:** Test main features in the whole WCS based on core flow
- **Method:** Black-box testing

## 2. Data Relationship Diagram

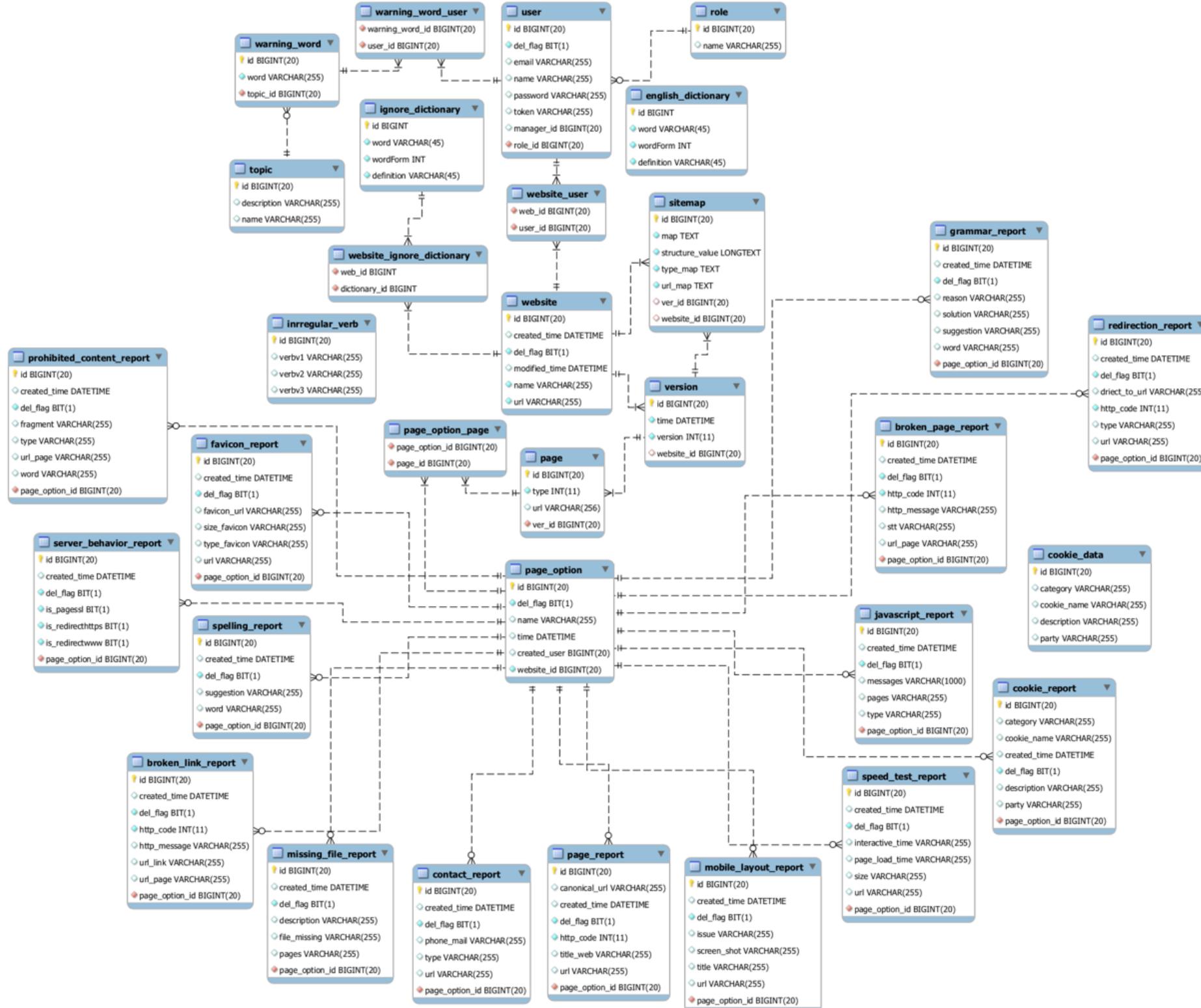


Figure 91 Physical Diagram

| <b>Entity Data Dictionary: Describe content of all tables</b> |                        |                                                                            |
|---------------------------------------------------------------|------------------------|----------------------------------------------------------------------------|
| <b>No.</b>                                                    | <b>Table Name</b>      | <b>Description</b>                                                         |
| 1                                                             | user                   | Contain account information.                                               |
| 2                                                             | roles                  | Contain role information.                                                  |
| 3                                                             | website                | Contain website information.                                               |
| 4                                                             | website_user           | Contain information about relationship between a user with a website.      |
| 5                                                             | sitemap                | Contain sitemap information.                                               |
| 6                                                             | version                | Contain version information.                                               |
| 7                                                             | page                   | Contain page information.                                                  |
| 8                                                             | page_option            | Contain page option information.                                           |
| 9                                                             | page_option_page       | Contain information about relationship between a page with a page option   |
| 10                                                            | warning_word           | Contain warning word information.                                          |
| 11                                                            | warning_word_user      | Contain information about relationship between a user with a warning word. |
| 12                                                            | topic                  | Contain topic information.                                                 |
| 13                                                            | spelling_report        | Contain spelling test report information.                                  |
| 14                                                            | speed_test_report      | Contain speed test report information.                                     |
| 15                                                            | server_behavior_report | Contain server behavior report information.                                |
| 16                                                            | cookie_report          | Contain cookie report information.                                         |
| 17                                                            | broken_link_report     | Contain broken link test report information.                               |
| 18                                                            | missing_file_report    | Contain missing file report information.                                   |
| 19                                                            | javascript_report      | Contain javascript report information.                                     |

|    |                           |                                                |
|----|---------------------------|------------------------------------------------|
| 20 | prohibited_content_report | Contain prohibited content report information. |
| 21 | mobile_layout_report      | Contain mobile layout report information.      |
| 22 | broken_page_report        | Contain broken page report information.        |
| 23 | page_report               | Contain page report information.               |
| 24 | redirection_report        | Contain redirection report information.        |
| 25 | favicon_report            | Contain favicon report information.            |
| 26 | grammar report            | Contain grammar report information.            |
| 27 | contact_report            | Contain contact report information.            |

Table 123 Physical Diagram Dictionary

### 3. Test Plan

The overall purpose of testing is to ensure WCS meets its entire technical, functional and business requirement. The purpose of this document is to describe the test plan and strategy for testing the WCS. The following part will describe which features to be tested and which will not.

#### 3.1. Features To Be Tested

- Staff:
  - Manage page option
  - Spelling test
  - Grammar test
  - Broken links test
  - Broken pages test
  - Missing files test
  - Prohibited content test
  - Mobile layout test
  - Speed test
  - JavaScript errors test
  - Favicons test
  - Cookie law test
  - Server behavior test
  - Pages test
  - Contact details test
  - Redirections test

### **3.2. Features Not To Be Tested**

- Guest: Login, sign up.
- Manager:
  - Manage Staff
  - Manage Website
  - Manager word list

## **4. System Testing Test Case**

### ***4.1 Staff Test Case***

#### ***4.1.1 Manage page option***

| <b>ID</b>    | <b>Test Case Description</b>                | <b>Test case procedure</b>                                                                                       | <b>Expected output</b>                               | <b><i>Inter Dependence -test Case</i></b> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|--------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|-------------------------------------------|---------------|------------------|-------------|
| <b>MPO_1</b> | Staff create a page option with blank name. | 1. Staff click "+" button.<br>2. Staff input blank.<br>3. Staff send create page option.                         | System shows error message "Please input something". | N/A                                       | Pass          | 26/11/2018       |             |
| <b>MPO_2</b> | Staff delete a page option.                 | 1. Staff double click to page option name<br>2. Staff send delete page option.                                   | Page option will be deleted.                         | N/A                                       | Pass          | 26/11/2018       |             |
| <b>MPO_3</b> | Staff select pages for page option.         | 1. Staff click Toggle to select or unselect page for page option.<br>2. Staff send select pages for page option. | Page option include pages selected will be created.  | N/A                                       | Pass          | 26/11/2018       |             |

Table 124 &lt;Test case&gt; Manage Page Option

#### 4.1.2 Broken links test

| <b>ID</b>    | <b>Test Case Description</b>                                      | <b>Test case procedure</b>                                     | <b>Expected output</b>                                                                                 | <b><i>Inter Dependence -test Case</i></b> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|--------------|-------------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|-------------------------------------------|---------------|------------------|-------------|
| <b>BLT_1</b> | Staff send broken links test command with page option unselected. | 1. Staff click “Check” button.                                 | System return test result with information: “Page”, “Broken Link”, “Issue” of root page.               | N/A                                       | Pass          | 26/11/2018       |             |
| <b>BLT_2</b> | Staff send broken links test command with page option selected.   | 1. Staff select page option.<br>2. Staff click “Check” button. | System return test result with information: “Page”, “Broken Link”, “Issue” of all page in page option. | N/A                                       | Pass          | 26/11/2018       |             |

Table 125&lt;Test case&gt; Broken Links Test

#### 4.1.3 Broken pages test

| <b>ID</b> | <b>Test Case Description</b> | <b>Test case procedure</b> | <b>Expected output</b> | <b><i>Inter Dependence -test Case</i></b> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|-----------|------------------------------|----------------------------|------------------------|-------------------------------------------|---------------|------------------|-------------|
|-----------|------------------------------|----------------------------|------------------------|-------------------------------------------|---------------|------------------|-------------|

|              |                                                                   |                                                                |                                                                                                   |     |      |            |  |
|--------------|-------------------------------------------------------------------|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------|-----|------|------------|--|
| <b>BPT_1</b> | Staff send broken pages test command with page option unselected. | 1. Staff click "Check" button.                                 | System return test result with information: "Page", "Status", "Issue" of root page.               | N/A | Pass | 26/11/2018 |  |
| <b>BPT_2</b> | Staff send broken pages test command with page option selected.   | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Page", "Status", "Issue" of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table 126 &lt;Test case&gt; Broken Pages Test

#### 4.1.4 Prohibited content test

| <b>ID</b>    | <b>Test Case Description</b>                                            | <b>Test case procedure</b>     | <b>Expected output</b>                                                                         | <b><i>Inter Dependence -test Case</i></b> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|--------------|-------------------------------------------------------------------------|--------------------------------|------------------------------------------------------------------------------------------------|-------------------------------------------|---------------|------------------|-------------|
| <b>PCT_1</b> | Staff send prohibited content test command with page option unselected. | 1. Staff click "Check" button. | System return test result with information: "Word", "Type", "Fragment" and "URL" of root page. | N/A                                       | Pass          | 26/11/2018       |             |

|              |                                                                       |                                                                |                                                                                                           |     |      |            |  |
|--------------|-----------------------------------------------------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|-----|------|------------|--|
| <b>PCT_2</b> | Staff send prohibited content test command with page option selected. | 1. Staff select page option.<br>2. Staff click “Check” button. | System return test result with information: “Word”, “Type”, “Fragment”, “URL” of all page in page option. | N/A | Pass | 26/11/2018 |  |
|--------------|-----------------------------------------------------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|-----|------|------------|--|

Table 127 &lt;Test case&gt; Prohibited Content Test

#### 4.1.5 Mobile layout test

| <b>ID</b>    | <b>Test Case Description</b>                                       | <b>Test case procedure</b>                                     | <b>Expected output</b>                                                                                | <i>Inter Dependence -test Case</i> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|--------------|--------------------------------------------------------------------|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|------------------------------------|---------------|------------------|-------------|
| <b>MLT_1</b> | Staff send mobile layout test command with page option unselected. | 1. Staff click “Check” button.                                 | System return test result with information: “Screenshot”, “Page”, “Issue” of root page.               | N/A                                | Pass          | 26/11/2018       |             |
| <b>MLT_2</b> | Staff send mobile layout test command with page option selected.   | 1. Staff select page option.<br>2. Staff click “Check” button. | System return test result with information: “Screenshot”, “Page”, “Issue” of all page in page option. | N/A                                | Pass          | 26/11/2018       |             |

Table 128 &lt;Test case&gt; Mobile Layout Test

#### 4.1.6 Speed test

| ID          | Test Case Description                                      | Test case procedure                                            | Expected output                                                                                                         | <i>Inter Dependence - test Case</i> | Result | Test Date  | Note |
|-------------|------------------------------------------------------------|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------|--------|------------|------|
| <b>ST_1</b> | Staff send speed test command with page option unselected. | 1. Staff click "Check" button.                                 | System return test result with information: "Page", "Interactive time", "Load time", "Size" of root page.               | N/A                                 | Pass   | 26/11/2018 |      |
| <b>ST_2</b> | Staff send speed test command with page option selected.   | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Page", "Interactive time", "Load time", "Size" of all page in page option. | N/A                                 | Pass   | 26/11/2018 |      |

Table 129 &lt;Test case&gt; Speed Test

#### 4.1.7 JavaScript test

| ID           | Test Case Description                                           | Test case procedure                                            | Expected output                                                                                         | Inter Dependence - test Case | Result | Test Date  | Note |
|--------------|-----------------------------------------------------------------|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|------------------------------|--------|------------|------|
| <i>JST_1</i> | Staff send JavaScript test command with page option unselected. | 1. Staff click “Check” button.                                 | System return test result with information: “Error message”, “Type”, “Page” of root page.               | N/A                          | Pass   | 26/11/2018 |      |
| <i>JST_2</i> | Staff send JavaScript test command with page option selected.   | 1. Staff select page option.<br>2. Staff click “Check” button. | System return test result with information: “Error message”, “Type”, “Page” of all page in page option. | N/A                          | Pass   | 26/11/2018 |      |

Table 130 &lt;Test case&gt; Javascript Errors Test

#### 4.1.8 Cookie law test

| <b>ID</b>           | <b>Test Case Description</b>                                    | <b>Test case procedure</b>                                     | <b>Expected output</b>                                                                                             | <b><i>Inter Dependence -test Case</i></b> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|---------------------|-----------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------|---------------|------------------|-------------|
| <b><i>CLT_1</i></b> | Staff send cookie law test command with page option unselected. | 1. Staff click "Check" button.                                 | System return test result with information: "Name", "Category", "Party" and "Description" of root page.            | N/A                                       | Pass          | 26/11/2018       |             |
| <b><i>CLT_2</i></b> | Staff send cookie law test command with page option selected.   | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Name", "Category", "Party", "Description" of all page in page option. | N/A                                       | Pass          | 26/11/2018       |             |

Table 131 &lt;Test case&gt; Cookie Law Test

#### **4.1.9 Server behavior test**

| <b>ID</b>           | <b>Test Case Description</b>                 | <b>Test case procedure</b>     | <b>Expected output</b>                  | <b><i>Inter Dependence -test Case</i></b> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|---------------------|----------------------------------------------|--------------------------------|-----------------------------------------|-------------------------------------------|---------------|------------------|-------------|
| <b><i>SBT_1</i></b> | Staff send server behavior test command with | 1. Staff click "Check" button. | System return test result of root page. | N/A                                       | Pass          | 26/11/2018       |             |

|              |                                                                    |                                                                |                                                       |     |      |            |  |
|--------------|--------------------------------------------------------------------|----------------------------------------------------------------|-------------------------------------------------------|-----|------|------------|--|
|              | page option unselected.                                            |                                                                |                                                       |     |      |            |  |
| <b>SBT_2</b> | Staff send server behavior test command with page option selected. | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table 132 &lt;Test case&gt; Server Behavior Test

#### **4.1.10 Missing File test**

| <b>ID</b>    | <b>Test Case Description</b>                                      | <b>Test case procedure</b>     | <b>Expected output</b>                                                                                 | <i><b>Inter Dependence -test Case</b></i> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|--------------|-------------------------------------------------------------------|--------------------------------|--------------------------------------------------------------------------------------------------------|-------------------------------------------|---------------|------------------|-------------|
| <b>MFT_1</b> | Staff send missing file test command with page option unselected. | 1. Staff click "Check" button. | System return test result with information: "Files", "Description", "Pages" and "Action" of root page. | N/A                                       | Pass          | 26/11/2018       |             |

|              |                                                                                                    |                                                                                                |                                                                                                                                        |     |      |            |  |
|--------------|----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-----|------|------------|--|
| <b>MFT_2</b> | Staff send missing file test command with page option selected.                                    | 1. Staff select page option.<br>2. Staff click "Check" button.                                 | System return test result with information: "Files", "Description", "Pages" and "Action" of all page in page option.                   | N/A | Pass | 26/11/2018 |  |
| <b>MFT_3</b> | Staff send missing file test command with page option unselected and select specific type to test. | 1. Staff select type to test.<br>2. Staff click "Check" button.                                | System return test result with information: "Files", "Description", "Pages" and "Action" of root page and type selected.               | N/A | Pass | 26/11/2018 |  |
| <b>MFT_3</b> | Staff send missing file test command with page option selected and select specific type to test.   | 1. Staff select page option.<br>2. Staff select type to test.<br>3. Staff click "Check" button | System return test result with information: "Files", "Description", "Pages" and "Action" of all page in page option and type selected. | N/A | Pass | 26/11/2018 |  |

Table 133 &lt;Test case&gt; Missing File Test

#### 4.1.11 Favicon test

| ID    | Test Case Description | Test case procedure | Expected output | Inter Dependence -test Case | Result | Test Date | Note |
|-------|-----------------------|---------------------|-----------------|-----------------------------|--------|-----------|------|
| <hr/> |                       |                     |                 |                             |        |           |      |

|              |                                                               |                                                                |                                                                                                                       |     |      |            |  |
|--------------|---------------------------------------------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|-----|------|------------|--|
| <b>CLT_1</b> | Staff send favicons test command with page option unselected. | 1. Staff click "Check" button.                                 | System return test result with information: "Favicon", "URL", "Width", "Type" and "Pages" of root page.               | N/A | Pass | 26/11/2018 |  |
| <b>CLT_2</b> | Staff send favicons test command with page option selected.   | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Favicon", "URL", "Width", "Type" and "Pages" of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table 134 &lt;Test case&gt; Favicon Test

#### 4.1.12 Contact Details test

| ID           | Test Case Description                                                | Test case procedure            | Expected output                                                                                   | Inter Dependence -test Case | Result | Test Date  | Note |
|--------------|----------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------|-----------------------------|--------|------------|------|
| <b>CDT_1</b> | Staff send contact details test command with page option unselected. | 1. Staff click "Check" button. | System return test result with information: "Contact", "Page affected" and "Action" of root page. | N/A                         | Pass   | 26/11/2018 |      |

|              |                                                                    |                                                                |                                                                                                                 |     |      |            |  |
|--------------|--------------------------------------------------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-----|------|------------|--|
| <b>CDT_2</b> | Staff send contact details test command with page option selected. | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Contact", "Page affected" and "Action" of all page in page option. | N/A | Pass | 26/11/2018 |  |
|--------------|--------------------------------------------------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-----|------|------------|--|

Table 135 &lt;Test case&gt; Contact Details Test

#### 4.1.14 Redirection test

| ID           | Test Case Description                                            | Test case procedure                                            | Expected output                                                                                                        | Inter Dependence -test Case | Result | Test Date  | Note |
|--------------|------------------------------------------------------------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------|------------|------|
| <b>RET_1</b> | Staff send redirection test command with page option unselected. | 1. Staff click "Check" button.                                 | System return test result with information: "Web Address", "Directs to", "Type" and "Code" of root page.               | N/A                         | Pass   | 26/11/2018 |      |
| <b>RET_2</b> | Staff send redirection test command with page option selected.   | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Web Address", "Directs to", "Type" and "Code" of all page in page option. | N/A                         | Pass   | 26/11/2018 |      |

Table 136 &lt;Test case&gt; Redirection Test

#### 4.1.14 Pages test

| <b>ID</b>    | <b>Test Case Description</b>                               | <b>Test case procedure</b>                                     | <b>Expected output</b>                                                                                                     | <i>Inter Dependence -test Case</i> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|--------------|------------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|------------------------------------|---------------|------------------|-------------|
| <b>CDT_1</b> | Staff send pages test command with page option unselected. | 1. Staff click "Check" button.                                 | System return test result with information: "Title", "Web Address", "Canonical URL" and "HTTP" of root page.               | N/A                                | Pass          | 26/11/2018       |             |
| <b>CDT_2</b> | Staff send pages test command with page option selected.   | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Title", "Web Address", "Canonical URL" and "HTTP" of all page in page option. | N/A                                | Pass          | 26/11/2018       |             |

Table 137 &lt;Test case&gt; Pages Test

#### 4.1.15 Spelling test

| <b>ID</b>    | <b>Test Case Description</b>                                  | <b>Test case procedure</b>     | <b>Expected output</b>                                             | <i>Inter Dependence -test Case</i> | <b>Result</b> | <b>Test Date</b> | <b>Note</b> |
|--------------|---------------------------------------------------------------|--------------------------------|--------------------------------------------------------------------|------------------------------------|---------------|------------------|-------------|
| <b>SPT_1</b> | Staff send spelling test command with page option unselected. | 1. Staff click "Check" button. | System return test result with information: "Word", "Suggestions", | N/A                                |               |                  |             |

|              |                                                             |                                                                |                                                                                                                                |     |  |  |  |
|--------------|-------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|-----|--|--|--|
|              |                                                             |                                                                | "Found in", "Page" and "Action" of root page.                                                                                  |     |  |  |  |
| <b>SPT_2</b> | Staff send spelling test command with page option selected. | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Word", "Suggestions", "Found in", "Page" and "Action" of all page in page option. | N/A |  |  |  |

Table 138 &lt;Test case&gt; Spelling Test

**4.1.16 Grammar test**

| ID           | Test Case Description                                        | Test case procedure            | Expected output                                                                            | <i>Inter Dependence -test Case</i> | Result | Test Date | Note |
|--------------|--------------------------------------------------------------|--------------------------------|--------------------------------------------------------------------------------------------|------------------------------------|--------|-----------|------|
| <b>GRT_1</b> | Staff send grammar test command with page option unselected. | 1. Staff click "Check" button. | System return test result with information: "Excerpt", "Issue", and "Action" of root page. | N/A                                |        |           |      |

|              |                                                            |                                                                |                                                                                                          |     |  |  |  |
|--------------|------------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-----|--|--|--|
| <b>GRT_2</b> | Staff send grammar test command with page option selected. | 1. Staff select page option.<br>2. Staff click "Check" button. | System return test result with information: "Excerpt", "Issue", and "Action" of all page in page option. | N/A |  |  |  |
|--------------|------------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-----|--|--|--|

Table 139 &lt;Test case&gt; Grammar Test

## F. Software User's Manual

### 1. Installation Guide

#### 1.1 Setting up environment at server side

##### 1.1.1 Hardware requirements

| Server                     | Minimum Requirements                 | Recommended                                   |
|----------------------------|--------------------------------------|-----------------------------------------------|
| <b>Internet Connection</b> | Cable, Wi-Fi (8 Mbps)                | Cable, Wi-Fi (50 Mbps or more)                |
| <b>Operating System</b>    | XP, Vista, 7, 10, Window Server 2008 | 10, Window Server 2008                        |
| <b>Computer Processor</b>  | Intel® Xeon ® 1.4GHz                 | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| <b>Computer Memory</b>     | 4GB RAM                              | 32GB or more                                  |
| <b>Storage space</b>       | 1GB                                  | 5GB or more                                   |

Table 140 - Hardware requirement for server

##### 1.1.2 Software requirements

| Name                    | Name/Version                                      | Description                                     |
|-------------------------|---------------------------------------------------|-------------------------------------------------|
| <b>Environment</b>      | Java EE 8.0, Node v10, npm v6                     | Specification for developing web application    |
| <b>Operating System</b> | Window 10                                         | Operating system and platform for development   |
| <b>Modeling tool</b>    | StarUML                                           | Used to design diagram                          |
| <b>IDE</b>              | IntelliJ IDEA 2018.1.5, Visual Studio Code 1.27.2 | Programming tools                               |
| <b>DBMS</b>             | MySQL 8.0                                         | Used to create & manage the database for system |
| <b>Web server</b>       | Apache Tomcat 8                                   | Deployment environment                          |
| <b>Web browser</b>      | Chrome 69 or above.                               | Testing browser                                 |

Table 141 - Software requirement for server

## 1.2 Deployment at server side

### 1.2.1 Prepare deployment package

**Step 1:** Download and Install Java 8

Link download:

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html>

**Step 2:** Download and setup Tomcat 8

Link download: <https://tomcat.apache.org/download-80.cgi>

**Step 3:** Download and Install Node v10

Link download: <https://nodejs.org/en/download/>

**Step 4:** Download and Install IntelliJ IDE

Link download:

<https://www.jetbrains.com/idea/download/#section=windows>

**Step 5:** Download, Install and configure MySQL 8.0

Link download: <https://dev.mysql.com/downloads/installer/>

### 1.2.2 Configure server before deploy

Step 1: Go to %Tomcat folder%\conf\

Step 2: Edit port to 8080 in server.xml at

```
<Connector port="8080" protocol="HTTP/1.1"
 connectionTimeout="20000"
 redirectPort="8443" />
```

Step 3: Delete all files and folders in %Tomcat folder%\webapps

Step 4: Rename production java build to ROOT.WAR and copy to  
%Tomcat folder%\webapps.

## 1.3 Setting up environment at client side

The client devices should have one of the following browsers to access the website:

- Google Chrome

Link download: <https://www.google.com/chrome/>

- Firefox

Link download: <https://www.mozilla.org/en-US/firefox/channel/desktop/>

## 2. User Guide

### 2.1 Manager

#### 2.1.1 Add Staff

The screenshot shows a web application interface for managing staff. On the left, there's a sidebar with navigation links: Dashboard, Manage Staff (which is selected and highlighted in blue), Manage Website, and Manage Wordlist. The main content area is titled 'Manage Staff'. It displays a table with two rows of staff information. Each row includes the staff's name, websites assigned to them, and buttons for 'Edit' and 'Delete'. The top right corner of the interface shows a user profile (nguyen duc trinh) and a 'Logout' button.

| No | Information                      | Assign for Website                                                                     | Action                    |
|----|----------------------------------|----------------------------------------------------------------------------------------|---------------------------|
| 1  | Name: trinh<br>Username: trinh   | http://genkvn<br>http://gamek.vn<br>http://topsportsmansguide.com<br>http://fpt.edu.vn | <b>Edit</b> <b>Delete</b> |
| 2  | Name: trinh1<br>Username: trinh1 | http://genkvn<br>http://gamek.vn                                                       | <b>Edit</b> <b>Delete</b> |

Figure 92 - Add Staff UI

The screenshot shows a modal dialog titled 'Add Staff'. It contains three input fields: 'Staff Name' with the value 'trinh' (labeled 3), 'Staff UserName' with the value 'nguyentrinh' (labeled 4), and 'Staff password' with the value '\*\*\*\*\*' (labeled 5). At the bottom of the dialog are 'Cancel' and 'Done' buttons, with the 'Done' button being highlighted (labeled 6).

Figure 93 - Add Staff Popup

| Step          | Description                                |
|---------------|--------------------------------------------|
| Pre-condition | Login as Manager                           |
| 1             | Click “Manage Staff” in the navigation bar |
| 2             | Click “Add” button                         |
| 3             | Fill the field “Staff Name”                |
| 4             | Fill the field “Staff Username”            |

|   |                                 |
|---|---------------------------------|
| 5 | Fill the field “Staff Password” |
| 6 | Click “Done” button             |

Table 142 - Add Staff Steps

### 2.1.2 Edit Staff

The screenshot shows a web application interface titled "Manage Staff". On the left, there is a vertical navigation bar with options: Dashboard, Manage Staff (highlighted with a red box labeled '1'), Manage Website, and Manage Wordlist. The main content area is titled "Manage Staff" and displays a table of staff members. The table has columns for No, Information, Assign for Website, and Action. Two staff entries are listed: one with Name: trinh and Username: trinh, and another with Name: trinh1 and Username: trinh1. Each entry has an "Edit" button (orange) and a "Delete" button (red). At the top right of the content area, there are "Add" and "Search..." buttons.

Figure 94 - Edit Staff UI

The screenshot shows an "Edit Staff" dialog box. It contains three input fields: "Staff Name" (with value "trinh"), "Staff UserName" (with value "trinh"), and "Staff password (Leave blank if no change)" (with value "\*\*\*\*\*"). Below these fields is a section titled "Website Assigned" listing several URLs. At the bottom right of the dialog are "Cancel" and "Done" buttons, with a red box labeled '6' over the "Done" button.

Figure 95 - Edit Staff Popup

| Step          | Description                                |
|---------------|--------------------------------------------|
| Pre-condition | Login as Manager                           |
| 1             | Click “Manage Staff” in the navigation bar |
| 2             | Click “Edit” button                        |
| 3             | Fill the field “Staff Name”                |
| 4             | Fill the field “Staff Username”            |
| 5             | Fill the field “Staff Password”            |
| 6             | Click “Done” button                        |

Table 143 - Edit Staff Steps

### 2.1.3 Delete Staff

| No | Information                      | Assign for Website                                                                     | Action                                      |
|----|----------------------------------|----------------------------------------------------------------------------------------|---------------------------------------------|
| 1  | Name: trinh<br>Username: trinh   | http://genkvn<br>http://gamenekvn<br>https://www.sportmansguide.com<br>http://ftpeduvn | <a href="#">Edit</a> <a href="#">Delete</a> |
| 2  | Name: trinh1<br>Username: trinh1 | http://genkvn<br>http://gamenekvn                                                      | <a href="#">Edit</a> <a href="#">Delete</a> |

Figure 96 - Delete Staff UI

### Delete Your Staff

Are you sure to staff this website ?

No [Yes](#) **3**

Figure 97 - Delete Staff Popup

| Step          | Description                                |
|---------------|--------------------------------------------|
| Pre-condition | Login as Manager                           |
| 1             | Click “Manage Staff” in the navigation bar |
| 2             | Click “Delete” button                      |
| 3             | Click “Yes” button                         |

Table 144 - Delete Staff Steps

### 2.1.4 Make new version of the website

| No | Name           | URL                            | Version | Lastest updated | New version            | Action                                                                                          |
|----|----------------|--------------------------------|---------|-----------------|------------------------|-------------------------------------------------------------------------------------------------|
| 1  | genk           | http://genkvn                  | 3       | 1/12/2018       | <a href="#">2 Make</a> | <a href="#">View Sitemap</a> <a href="#">Assign</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 2  | gamenek        | http://gamenekvn               | 1       | 22/11/2018      | <a href="#">Make</a>   | <a href="#">View Sitemap</a> <a href="#">Assign</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 3  | Sportmansguide | https://www.sportmansguide.com | 1       | 22/11/2018      | <a href="#">Make</a>   | <a href="#">View Sitemap</a> <a href="#">Assign</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 4  | ftpt           | http://ftpeduvn                | 1       | 24/11/2018      | <a href="#">Make</a>   | <a href="#">View Sitemap</a> <a href="#">Assign</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 5  | qwe            | qwe                            | 1       | 27/11/2018      | <a href="#">Make</a>   | <a href="#">View Sitemap</a> <a href="#">Assign</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 6  | dsd            | https://insites.com            | 1       | 27/11/2018      | <a href="#">Make</a>   | <a href="#">View Sitemap</a> <a href="#">Assign</a> <a href="#">Edit</a> <a href="#">Delete</a> |

Figure 98 - Make new version UI

| Step          | Description                                  |
|---------------|----------------------------------------------|
| Pre-condition | Login as Manager                             |
| 1             | Click “Manage Website” in the navigation bar |
| 2             | Click “Make” button                          |

Table 145 - Make new version Steps

### 2.1.5 View Sitemap

The screenshot shows a web-based management system for websites. On the left, there's a sidebar with navigation links: Dashboard, Manage Staff, Manage Website, and Manage Wordlist. The main area is titled 'Manage Website'. A red number '1' is placed above the table header. The table lists six websites with columns for No., Name, URL, Version, Last updated, New version, and Action. The 'Action' column contains buttons for 'Make', 'View Sitemap' (highlighted with a red '2'), 'Assign', 'Edit', and 'Delete'. The 'View Sitemap' button is blue with white text.

| No | Name           | URL                            | Version | Last updated | New version           | Action                                                                                                |
|----|----------------|--------------------------------|---------|--------------|-----------------------|-------------------------------------------------------------------------------------------------------|
| 1  | genk           | http://genk.vn                 | 3       | 1/12/2018    | <button>Make</button> | <button>2 View Sitemap</button> <button>Assign</button> <button>Edit</button> <button>Delete</button> |
| 2  | gamek          | http://gamek.vn                | 1       | 22/11/2018   | <button>Make</button> | <button>View Sitemap</button> <button>Assign</button> <button>Edit</button> <button>Delete</button>   |
| 3  | Sportmansguide | https://www.sportmansguide.com | 1       | 22/11/2018   | <button>Make</button> | <button>View Sitemap</button> <button>Assign</button> <button>Edit</button> <button>Delete</button>   |
| 4  | fpt            | http://fpt.edu.vn              | 1       | 24/11/2018   | <button>Make</button> | <button>View Sitemap</button> <button>Assign</button> <button>Edit</button> <button>Delete</button>   |
| 5  | qwe            | qwe                            | 1       | 27/11/2018   | <button>Make</button> | <button>View Sitemap</button> <button>Assign</button> <button>Edit</button> <button>Delete</button>   |
| 6  | dsd            | https://insites.com            | 1       | 27/11/2018   | <button>Make</button> | <button>View Sitemap</button> <button>Assign</button> <button>Edit</button> <button>Delete</button>   |

Figure 99 - View Sitemap UI

Visual Sitemap of Website: Tiki - Tuyen dung

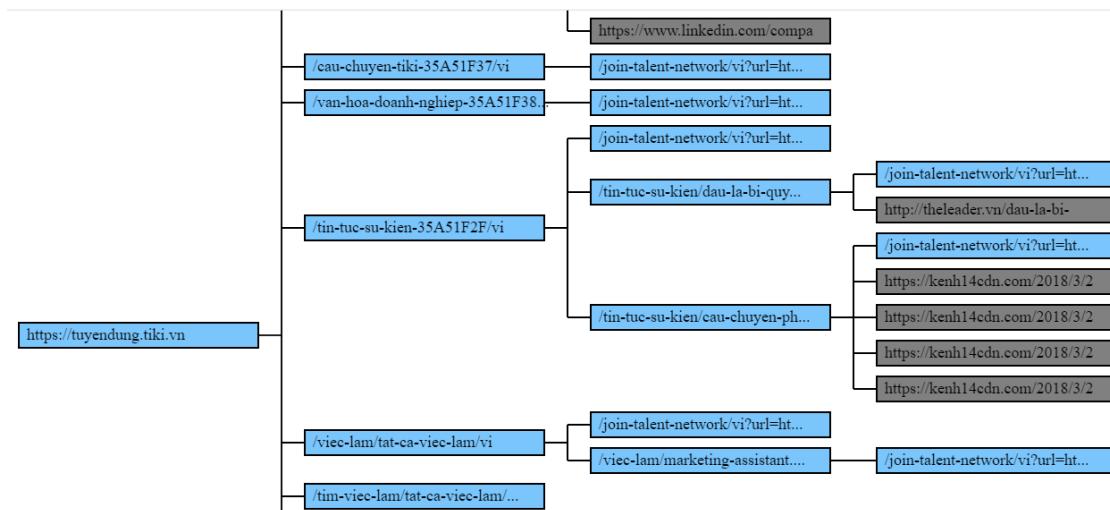


Figure 100 - Sitemap Screen

| Step          | Description                                  |
|---------------|----------------------------------------------|
| Pre-condition | Login as Manager                             |
| 1             | Click "Manage Website" in the navigation bar |
| 2             | Click "View Sitemap" button                  |

Table 146 - View Sitemap Steps

### 2.1.6 Assign website for Staff

| No | Name            | URL                             | Version | Last updated | New version           | Action                                                                                                                                 |
|----|-----------------|---------------------------------|---------|--------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 1  | genk            | http://genk.vn                  | 3       | 1/12/2018    | <button>Make</button> | <button>View Sitemap</button> <span style="color: red;">2</span> <button>Assign</button> <button>Edit</button> <button>Delete</button> |
| 2  | gamek           | http://gamek.vn                 | 1       | 22/11/2018   | <button>Make</button> | <button>View Sitemap</button> <span style="color: red;">2</span> <button>Assign</button> <button>Edit</button> <button>Delete</button> |
| 3  | Sportsmansguide | https://www.sportsmansguide.com | 1       | 22/11/2018   | <button>Make</button> | <button>View Sitemap</button> <span style="color: red;">2</span> <button>Assign</button> <button>Edit</button> <button>Delete</button> |
| 4  | fpt             | http://fpt.edu.vn               | 1       | 24/11/2018   | <button>Make</button> | <button>View Sitemap</button> <span style="color: red;">2</span> <button>Assign</button> <button>Edit</button> <button>Delete</button> |
| 5  | qwe             | qwe                             | 1       | 27/11/2018   | <button>Make</button> | <button>View Sitemap</button> <span style="color: red;">2</span> <button>Assign</button> <button>Edit</button> <button>Delete</button> |
| 6  | dstd            | https://insites.com             | 1       | 27/11/2018   | <button>Make</button> | <button>View Sitemap</button> <span style="color: red;">2</span> <button>Assign</button> <button>Edit</button> <button>Delete</button> |

Figure 101 - Assign website UI

Assign Website

Staffs:

- Staff: trinh - trinh ✖
- trinh1

Figure 102 - Assign website popup

Assign Website

Staffs:

- Staff: trinh - trinh ✖
- Staff: trinh1 - trinh1 ✖

Cancel 5 Done

Figure 103 - Assign website popup

| Step          | Description                                  |
|---------------|----------------------------------------------|
| Pre-condition | Login as Manager                             |
| 1             | Click “Manage Website” in the navigation bar |
| 2             | Click “Assign” button                        |
| 3             | Click “Staffs” dropdown                      |
| 4             | Select Staff                                 |
| 5             | Click “Done” button                          |

Table 147 - Assign website steps

### 2.1.7 Add warning word

| No | Word     | Lastest updated | Type                  | Action                                        |
|----|----------|-----------------|-----------------------|-----------------------------------------------|
| 1  | gun      | 2018-11-22      | Weapons               | <button>Edit</button> <button>Delete</button> |
| 2  | rifle    | 2018-11-22      | Weapons               | <button>Edit</button> <button>Delete</button> |
| 3  | shotguns | 2018-11-22      | Weapons               | <button>Edit</button> <button>Delete</button> |
| 4  | naked    | 2018-11-22      | Sexual                | <button>Edit</button> <button>Delete</button> |
| 5  | poop     | 2018-11-22      | Potentially offensive | <button>Edit</button> <button>Delete</button> |
| 6  | ak       | 2018-11-23      | Weapons               | <button>Edit</button> <button>Delete</button> |
| 7  | Funix    | 2018-11-24      | Potentially offensive | <button>Edit</button> <button>Delete</button> |
| 8  | FPT      | 2018-11-24      | Potentially offensive | <button>Edit</button> <button>Delete</button> |
| 9  | bed      | 2018-11-24      | Potentially offensive | <button>Edit</button> <button>Delete</button> |

Figure 104 - Add warning word UI



Figure 105 - Add warning word popup

The screenshot shows the 'Add Word' modal with the following values: 'Word' field contains 'gun' (red number '3'), 'Type' dropdown is set to 'Weapons' (red number '4'). In the bottom right corner, there are 'Cancel' and 'Done' buttons, with a red number '6' next to the 'Done' button.

Figure 106 - Add warning word popup

| Step          | Description                                   |
|---------------|-----------------------------------------------|
| Pre-condition | Login as Manager                              |
| 1             | Click "Manage Wordlist" in the navigation bar |
| 2             | Click "Add" button                            |
| 3             | Fill the field "Word"                         |
| 4             | Click "Type" dropdown                         |
| 5             | Select Type                                   |
| 6             | Click "Done" button                           |

Table 148 - Add warning word steps

### 2.1.8 Edit warning word

| No | Word     | Lastest updated | Type                  | Action                     |
|----|----------|-----------------|-----------------------|----------------------------|
| 1  | gun      | 2018-11-22      | Weapons               | <span>2</span> Edit Delete |
| 2  | rifle    | 2018-11-22      | Weapons               | <span>2</span> Edit Delete |
| 3  | shotguns | 2018-11-22      | Weapons               | <span>2</span> Edit Delete |
| 4  | naked    | 2018-11-22      | Sexual                | <span>2</span> Edit Delete |
| 5  | poop     | 2018-11-22      | Potentially offensive | <span>2</span> Edit Delete |
| 6  | ak       | 2018-11-23      | Weapons               | <span>2</span> Edit Delete |
| 7  | Funix    | 2018-11-24      | Potentially offensive | <span>2</span> Edit Delete |
| 8  | FPT      | 2018-11-24      | Potentially offensive | <span>2</span> Edit Delete |
| 9  | bed      | 2018-11-24      | Potentially offensive | <span>2</span> Edit Delete |

Figure 107 - Edit warning word UI

Edit Word

Word

 3

Type

Weapons 4

Sexual 5

Potentially offensive

Figure 108 - Edit warning word popup

Edit Word

Word

Type

Weapons 6

Cancel Done

Figure 109 - Edit warning word popup

| Step          | Description                                   |
|---------------|-----------------------------------------------|
| Pre-condition | Login as Manager                              |
| 1             | Click "Manage Wordlist" in the navigation bar |

|   |                       |
|---|-----------------------|
| 2 | Click “Edit” button   |
| 3 | Fill the field “Word” |
| 4 | Click “Type” dropdown |
| 5 | Select Type           |
| 6 | Click “Done” button   |

Table 149 - Edit warning word steps

## 2.2 Staff

### 2.2.1 Select website to test

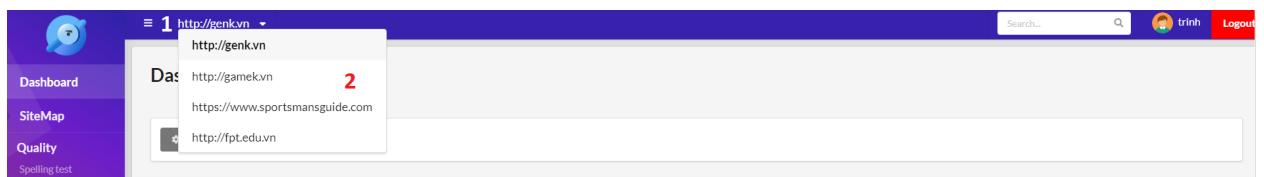


Figure 110 - Select website UI

| Step          | Description                            |
|---------------|----------------------------------------|
| Pre-condition | Login as Staff                         |
| 1             | Click “Website” dropdown in the header |
| 2             | Select website                         |

Table 150 - Select website steps

### 2.2.2 Add page option

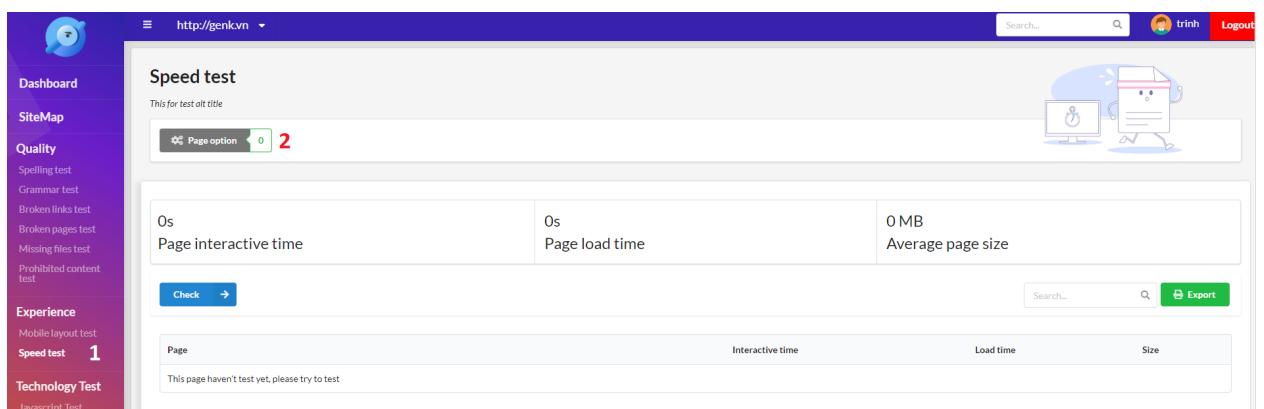


Figure 111 - Select page option UI

Page Option - <http://genk.vn>

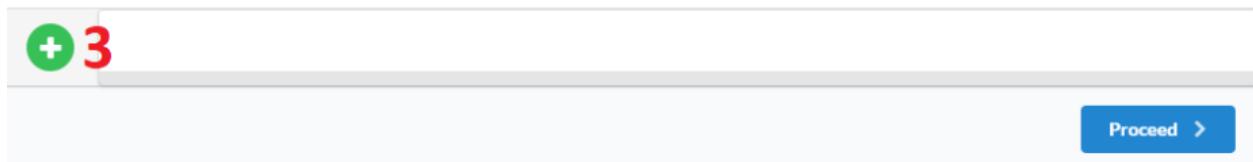


Figure 112 - Add page option popup

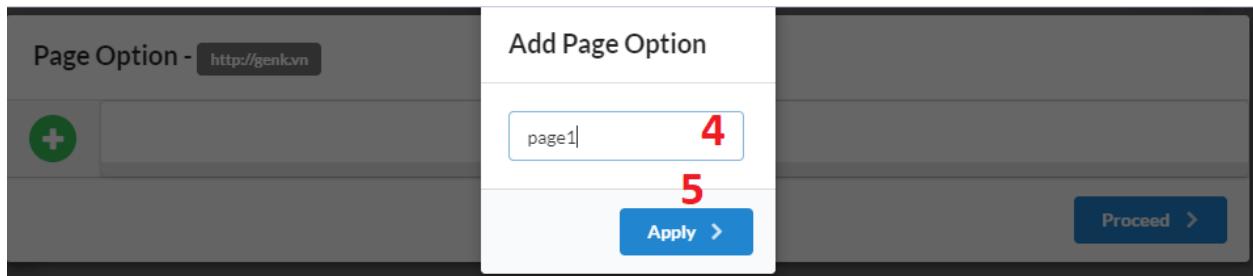


Figure 113 - Add page option popup

Page Option - <http://genk.vn>

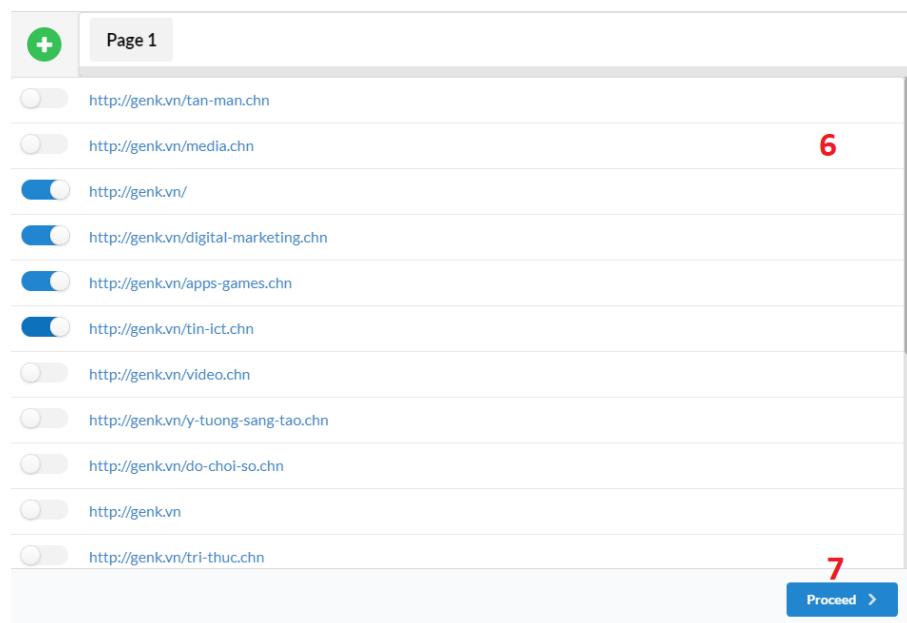


Figure 114 - Add page option popup

| Step          | Description                                    |
|---------------|------------------------------------------------|
| Pre-condition | Login as Staff                                 |
| 1             | Select any test function in the navigation bar |
| 2             | Click "Page Option" button                     |
| 3             | Click "Plus" button                            |

|   |                                   |
|---|-----------------------------------|
| 4 | Fill the field “Page option name” |
| 5 | Click “Apply” button              |
| 6 | Select page for page option       |
| 7 | Click “Proceed” button            |

Table 151 - Add page option steps

### 2.2.3 Rename or Delete page option

|                             |                      |                           |
|-----------------------------|----------------------|---------------------------|
| 0s<br>Page interactive time | 0s<br>Page load time | 0 MB<br>Average page size |
|-----------------------------|----------------------|---------------------------|

Figure 115 - Select page option UI

Figure 116 - Select page option popup

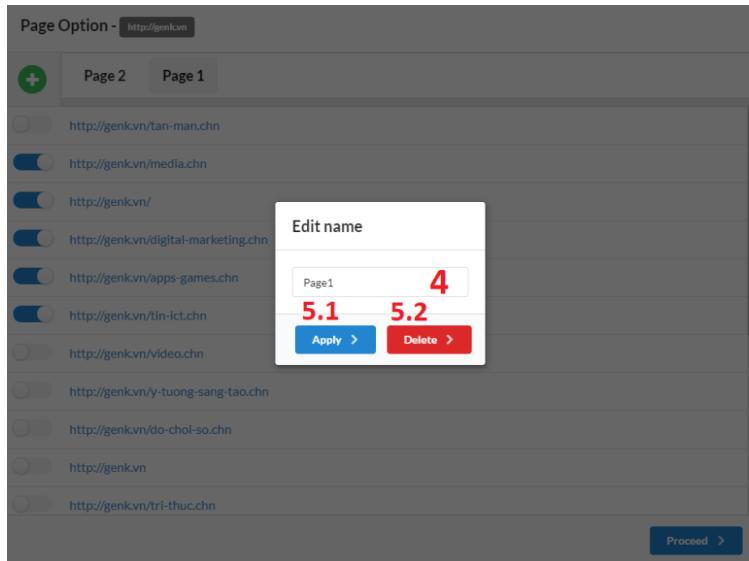


Figure 117 - Rename or Delete page option popup

| Step          | Description                                    |
|---------------|------------------------------------------------|
| Pre-condition | Login as Staff                                 |
| 1             | Select any test function in the navigation bar |
| 2             | Click "Page Option" button                     |
| 3             | Double click page option name                  |
| 4             | Fill the field "Page option name"              |
| 5.1           | Click "Apply" button to rename page option     |
| 5.2           | Click "Delete" button to delete page option    |

Table 152 - Rename or Delete page option steps

#### 2.2.4 Do a test function

The screenshot shows the 'Speed test' section of the website checker system. On the left sidebar, under 'Speed test', there is a count of 1. The main area displays performance metrics: 0s for Page interactive time and 0 MB for Average page size. Below these metrics is a table with columns for Page, Interactive time, Load time, and Size. A note at the bottom says 'This page haven't test yet, please try to test'.

Figure 118 - Select page option UI

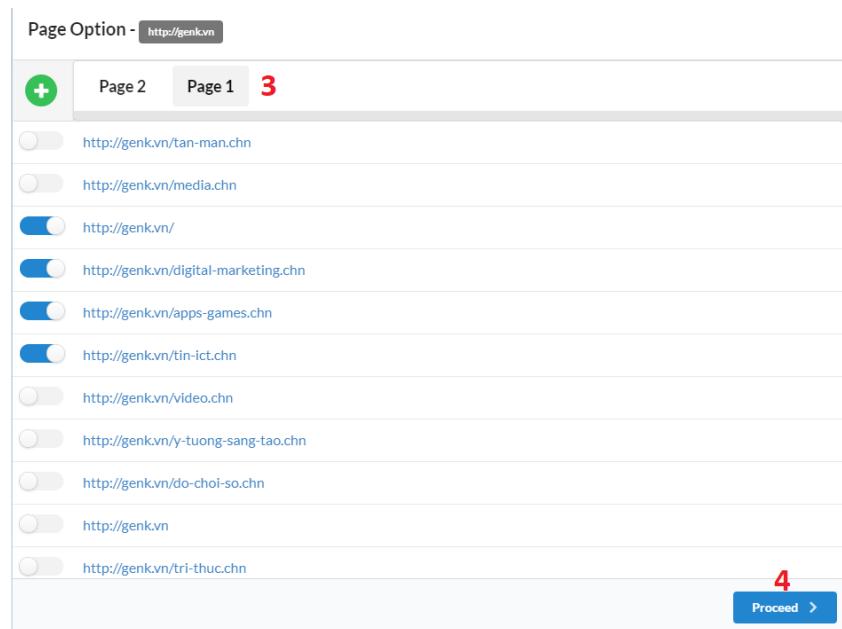


Figure 119 - Select page option popup

| Page                                 | Interactive time | Load time | Size   |
|--------------------------------------|------------------|-----------|--------|
| http://genk.vn/                      | 1.8s             | 4.6s      | 7.9 MB |
| http://genk.vn/digital-marketing.chn | 1.3s             | 4.6s      | 4.2 MB |
| http://genk.vn/tin-ict.chn           | 0.8s             | 4.0s      | 4.3 MB |
| http://genk.vn/apps-games.chn        | 2.3s             | 3.3s      | 2.5 MB |

Figure 120 - Test function UI

| Step          | Description                                    |
|---------------|------------------------------------------------|
| Pre-condition | Login as Staff                                 |
| 1             | Select any test function in the navigation bar |
| 2             | Click "Page Option" button                     |
| 3             | Click to choose page option                    |
| 4             | Click "Proceed" button                         |
| 5             | Click "Test" button                            |
| 6             | Click "Save" button to save result             |
| 7             | Click "Export" button to export result         |

Table 153 - Do a test function steps

## G. Appendix

1. **Spring Framework** [Online] Available: <https://spring.io/docs/reference>.
2. **The Unified Certification** [Online] Available: <https://www.omg.org/ocup-2/index.htm>.
3. **MySQL** [Online] Available: <https://dev.mysql.com/doc/>.
4. **ReactJS** [Online] Available: <https://reactjs.org/docs/getting-started.html>
5. **IntelliJ IDEA** [Online] Available: <https://www.jetbrains.com/idea/documentation/>
6. **Visual Studio Code** [Online] Available: <https://code.visualstudio.com/docs>
7. **Jsoup** [Online] Available: <https://jsoup.org/apidocs/overview-summary.html>
8. **Selenium** [Online] Available: <https://www.seleniumhq.org/docs/>
9. **Nodejs** [Online] Available: <https://nodejs.org/en/docs/>
10. **HttpCode** [Online] Available: <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
11. **Regular Expression** [Online] Available:  
<https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>
12. **Cookie Law** [Online] Available: <https://www.cookielaw.org/faq/>
13. **DFS and BFS Algorithms using Stacks and Queues, Professor Lawrence L. Larmore** [Online] Available:  
<http://web.cs.unlv.edu/larmore/Courses/CSC477/bfsDfs.pdf>
14. **The Breadth-First Search Algorithm, Khan Academy** [Online] Available:  
<https://www.khanacademy.org/computing/computer-science/algorithms/breadth-first-search/a/the-breadth-first-search-algorithm>
15. **Data Structure—Breadth First Traversal, TutorialsPoint** [Online]  
Available:  
[https://www.tutorialspoint.com/data\\_structures\\_algorithms/breadth\\_first\\_traversal.htm](https://www.tutorialspoint.com/data_structures_algorithms/breadth_first_traversal.htm)
16. **Binary tree: Level Order Traversal, mycodeschool** [Online] Available:  
<https://www.youtube.com/watch?v=86g8jAQug04>
17. **Breadth-First Traversal of a Tree, Computer Science Department of Boston University** [Online] Available:  
<https://www.cs.bu.edu/teaching/c/tree/breadth-first/>

**18. Digit-based sorting and data structures, Professor Avrim Blum [Online]**  
Available: <https://www.cs.cmu.edu/~avrim/451f11/recitations/rec0921.pdf>

**19. Lecture Notes on Tries, Professor Frank Pfenning [Online]** Available:  
<https://www.cs.cmu.edu/~fp/courses/15122-f10/lectures/18-tries.pdf>

**20. Algorithms: Tries, Robert Sedgewick and Kevin Wayne [Online]** Available:  
<https://algs4.cs.princeton.edu/lectures/52Tries.pdf>

**21. Tries, Brilliant Learning [Online]** Available: <https://brilliant.org/wiki/tries/>

**22. Tries, Daniel Ellard [Online]** Available:  
[http://ellard.org/dan/www/libsq/cb\\_1998/c06.pdf](http://ellard.org/dan/www/libsq/cb_1998/c06.pdf)

**23. Tries, Harvard CS50 [Online]** Available:  
<https://www.youtube.com/watch?v=TRg9DQFu0kU>

**24. Sitemap, Garenne Bigby [Online]** Available: <https://dynamapper.com/blog/16-sitemap-generator/164-top-50-sitemap-generators-for-creating-sitemaps>