**TDTP**

**Angelfire Website Creation Tool**

Academic year: 2023-2024, Spring Semester

Team Name: Audi TT (Testting Team)

Team members:

Forminte Andrei – Gabriel (332),

Guceanu George – Marian (933)

Contents

[1. Application Details. Investigated Features 3](#_Toc164118504)

[2. AC. IOs 3](#_Toc164118505)

[3. Testing Mission 3](#_Toc164118506)

[4. Testing Strategy 3](#_Toc164118507)

[5. Selected Test Design Techniques 3](#_Toc164118508)

[6. Test Design. Test implementation. Test execution. Test Report 4](#_Toc164118509)

[6.1. Test Design 4](#_Toc164118510)

[6.2. Test Implementation. Test Execution 4](#_Toc164118511)

[6.3. Test Report 5](#_Toc164118512)

[7. Issue Reporting 5](#_Toc164118513)

[8. Conclusions. Lessons Learned 5](#_Toc164118514)

# Application Details. Investigated Features

*Angelfire by Lycos is a pioneering web hosting service that allows users to create and publish their own websites for free. Established in the early days of the internet, Angelfire empowers individuals with limited technical knowledge to design and share their content online. With its user-friendly tools and customizable templates, Angelfire has been a go-to platform for personal websites, hobbyists, and small businesses looking to establish an online presence.*

*The following functionalities are tested by us, based on a couple of relevant factors, the biggest one being the user experience and the joy of use: website creation wizard, template customization, advertisement integration, basic user authentication and authorization, website publishing, multi-platform integration, time-performance and a few more.*

# AC. IOs

Application Context:

We are part of the testing team that joined early in the project, two months before product release.

We are junior testers and this is your first testing project.

**Part I:**

The project manager wants you to perform specification-based testing.

The product specifications are explicit and implicit as well.

**Part II:**

Check for hidden features and additional features that are not mentioned in the specifications.

Find as many as possible and test them.

Write a report on your findings and testing results.

***1. Register page***

***Inputs:***

***Member name***

***Password***

***Email address***

***Expected Outputs:***

***A validation error, or in case the fields are all correct, a success message***

***2. Login form***

***Inputs:***

***Username: design inconsistency***

***Password***

***Expected outputs:***

***Redirect to home page where user is logged in***

***3. Tour Our Website Builder button***

***Inputs:***

***Click on button***

***Expected outputs:***

***Website Builder presentation***

***4. View More Of Our Services button***

***Inputs:***

***Click on button***

***Expected outputs:***

***More Services presentation***

***5. Get Your Own Domain form***

***Inputs:***

***Domain name***

***Expected outputs:***

***Domain availability***

***6. Build A Website Now button***

***Inputs:***

***Click on button***

***Expected outputs:***

***Website builder when logged in or register page***

***7. Browse More Styles button***

***Inputs:***

***Click on button***

***Click on template***

***Expected outputs:***

***Page with template in detail***

***Placeholders***

***8. Advertisement section***

***ERR: section is empty***

***Inputs:***

***Click on button***

***Expected Outputs:***

***Advertisement page***

# Testing Mission

**Testing mission for Part I.**

For the first part, we went with a specification-based testing approach, where there are certain defined specifications, the explicit ones, and the ones that derive, mostly, from the previous ones, those being the implicit.

The mission of this testing part, as outlined in the provided information, seems to be twofold:

1. **Identify and validate explicit specifications**: The first mission is to ensure that the explicit specifications listed in the project requirements are met. This involves testing functionalities such as user registration and login, template selection and editing, website publishing, and PayPal integration.
2. **Uncover and address implicit specifications**: The second mission is to uncover and address implicit specifications that are not explicitly stated but are inferred or assumed to be required for the product. This includes testing aspects such as cross-browser compatibility, presentation page redirection, design consistency, page loading time, presence of advertisements, and website security.

**Testing mission for Part II.**

**Verify the intended purpose of the website**: The primary objective is to ensure that the website fulfills its intended purpose effectively. So, we will do the following:

1. **Identify hidden and additional features**: Another mission is to discover any hidden or additional features that are not explicitly mentioned in the specifications. This entails thorough exploration of the website to uncover any functionalities beyond what was initially outlined.
2. **Test the discovered additional features**: Once hidden or additional features are identified, the mission is to test them rigorously to ensure they function correctly and integrate seamlessly with the existing functionalities.
3. **Assess compatibility across browsers and platforms**: It's essential to check if the website functions consistently across various browsers and platforms, ensuring a seamless user experience for all visitors.
4. **Evaluate performance and address time-complexity issues**: Testing should include measuring the website's performance metrics such as loading time and page switching speed to identify any potential time-complexity-related issues that may impact user experience.
5. **Ensure website security**: Security testing is crucial to verify that the website adequately protects user data and credentials, utilizing features such as cookies, local storage, and proper credentials management to enhance security measures.
6. **Assess usability**: Usability testing aims to determine the website's ease of use, navigation, and understandability, ensuring it is user-friendly and intuitive for visitors.
7. **Validate reliability**: Finally, testing should focus on ensuring the website operates reliably without crashes, bugs, or errors, providing a seamless and consistent experience to users.

# Testing Strategy

**Testing strategy for Part I.**

For the first part, we moved along with specification-based testing, as it inquires the model-based strategy we kept in mind. Specification-based testing, also known as black-box testing, focuses on validating the behavior of a system based on its specifications or requirements. It involves designing test cases that directly correspond to the specified functionalities, inputs, outputs, and expected behavior outlined in the project requirements. The tester does not need to have knowledge of the internal code or implementation details of the system being tested; instead, they rely solely on the provided specifications to design and execute tests.

There are several reasons why specification-based testing is more suitable for the first part of your testing project:

1. **Alignment with provided specifications**: The first part of your project explicitly mentions using product specifications, both explicit and implicit, to guide the testing process. Specification-based testing directly aligns with this approach by focusing on validating the system against these specifications.
2. **Structured approach**: Specification-based testing provides a structured and systematic approach to testing, as test cases are derived directly from the documented requirements. This ensures comprehensive coverage of specified functionalities and helps in verifying that the system behaves as expected.
3. **Clear verification of requirements**: By directly testing against the specified requirements, specification-based testing provides clear verification of whether the system meets the expected criteria. This makes it easier to identify any deviations or discrepancies between the desired behavior and the actual implementation.
4. **Documentation and traceability**: Specification-based testing facilitates documentation and traceability, as test cases are explicitly linked to the documented requirements. This allows for easier tracking of test coverage and ensures that all specified functionalities are adequately tested.
5. **Reduced reliance on implementation details**: Since specification-based testing focuses solely on the externally observable behavior of the system, it reduces the reliance on internal implementation details. This makes it easier to maintain test cases and ensures that tests remain relevant even if the underlying implementation changes.

Overall, specification-based testing provides a structured and effective approach to validating system behavior against specified requirements, making it well-suited for the objectives outlined in the first part of your testing project.

**Testing strategy for Part II.**

We decided that exploratory testing is suitable for the second part, as it aligns with the analytical nature of it. Exploratory testing is an approach where testers explore the system, learn about its functionalities, and simultaneously design and execute test cases based on their observations and intuition. Unlike scripted testing, where test cases are predefined, exploratory testing allows testers to adapt their testing approach dynamically based on what they discover during testing.

Here's why exploratory testing would be a good choice for the second part:

1. **Flexibility and adaptability**: Exploratory testing is highly flexible and adaptable, making it well-suited for uncovering hidden or additional features that may not have been explicitly documented. Testers can explore the system freely, allowing them to discover functionalities that may not have been previously considered.
2. **Ability to uncover unexpected behaviors**: Since exploratory testing encourages testers to explore the system without predefined scripts, it enables them to uncover unexpected behaviors or functionalities that may not align with the documented specifications. This allows for a more thorough exploration of the system's capabilities.
3. **Efficient for testing unstructured areas**: In cases where the system's functionality is not fully defined or documented, exploratory testing can be particularly effective. Testers can use their creativity and intuition to explore unstructured areas of the system and identify potential issues or opportunities for improvement.
4. **Opportunity for creativity and innovation**: Exploratory testing allows testers to exercise their creativity and innovation by approaching testing with an open mind. This can lead to the discovery of innovative solutions, as well as uncovering unique features or functionalities that enhance the overall quality of the product.
5. **Encourages collaboration and knowledge sharing**: Since exploratory testing involves active exploration and learning, it encourages collaboration among testers and facilitates knowledge sharing. Testers can share their findings and insights, leading to a more comprehensive understanding of the system's behavior.

Overall, exploratory testing is well-suited for the objectives outlined in the second part of your testing project, as it provides the flexibility, adaptability, and creativity required to uncover hidden features and thoroughly test the system beyond the documented specifications.

# Selected Test Design Techniques

To address the acceptance criteria (AC) and input/output (IO) combinations covered by the provided test cases, various test design techniques are employed. Equivalence Partitioning and Boundary Value Analysis are utilized to test valid and invalid input combinations for the Register page, ensuring proper handling of boundary conditions. State Transition Testing is applied to verify correct transitions between logged-out and logged-in states for the Login form. Decision Table Testing helps cover different combinations of inputs and expected outputs for features like the Tour Our Website Builder button and View More Of Our Services button. Error Guessing is used to anticipate and test error-handling issues, particularly for inputs like the Domain name on the Get Your Own Domain form. Pairwise Testing efficiently covers different combinations of inputs, such as those involved in the Browse More Styles button. These techniques ensure comprehensive testing, addressing various test attributes and dimensions to uncover potential issues in the system's behavior and functionality.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **Test Strategy** | **Test Design Technique** | **~~Test Attributes~~** | **Dimension covered** | **Students and Features** |
| Part I | Model-based | Black box | Scenario-based | Coverage | Guceanu George – Marian Features: Domain Checker, Login |
|  | Model-based | Black box | Scenario-based | Activity | Forminte Andrei  Features: Register Page, Website Builder |
| Part 2 | Exploratory | Session-based testing | Requirements-based | Both of the above | All features and both students. |
|  |  |  |  |  |  |

# Test Design. Test implementation. Test execution. Test Report

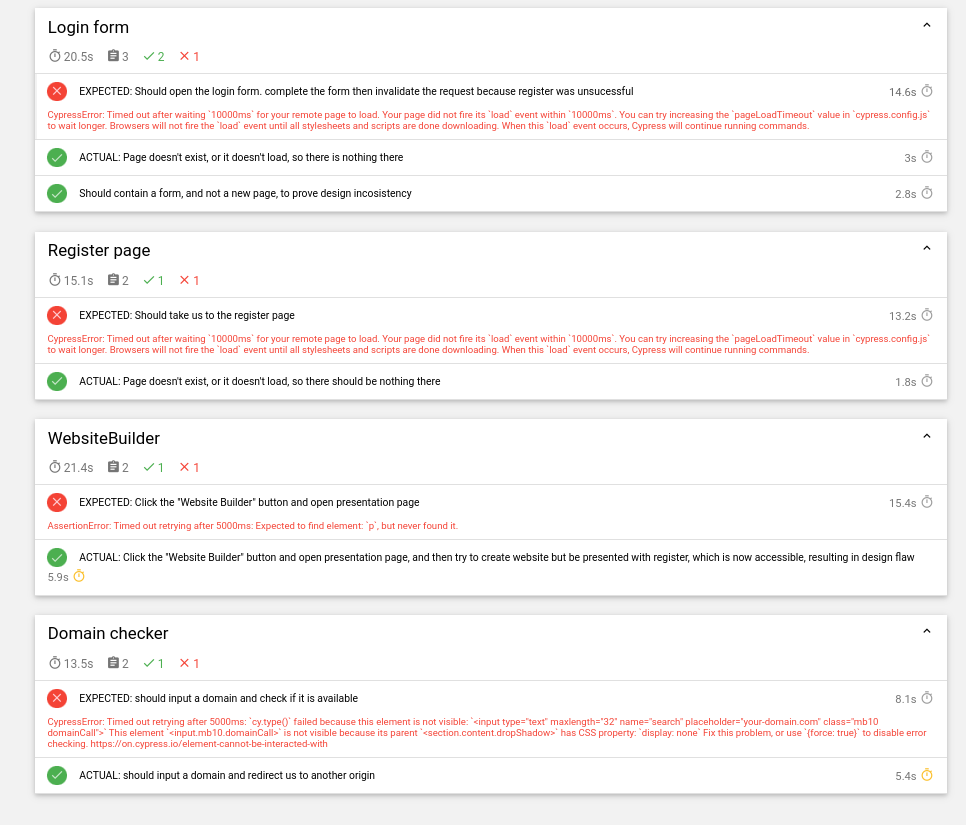
# Test Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Student** | **Feature(s)** | **Test Design Technique** | **Details** | **Input, Expected output** |
| Guceanu George – Marian | Domain Checker, Login | Scenario-based (Black Box) | Model-based approach, focusing on user scenarios to test coverage for Domain Checker and Login features | **Domain Checker**: Input: Domain name Expected Output: Domain availability or error message **Login**: Input: Username, Password Expected Output: Redirect to home page, user logged in |
| Forminte Andrei | Register Page, Website Builder | Scenario-based (Black Box) | Model-based approach, focusing on user scenarios to test activity for Register Page and Website Builder features | **Register Page**: Input: Member name, Password, Email address Expected Output: Validation error or success message **Website Builder**: Input: Click on "Tour Our Website Builder" button Expected Output: Website Builder presentation |
| Guceanu George – Marian | All features | Session-based (Exploratory) | Exploratory testing sessions to uncover hidden and additional features, using a requirements-based approach | **All features**: Inputs: Various based on features Expected Outputs: Functionality as per requirements, uncovering hidden issues |
| Forminte Andrei | All features | Session-based (Exploratory) | Exploratory testing sessions to uncover hidden and additional features, using a requirements-based approach | **All features**: Inputs: Various based on featuresExpected Outputs: Functionality as per requirements, uncovering hidden issues |
|  |  |  |  |  |

# Test Implementation. Test Execution

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Student** | **Feature(s)** | **Input, Expected Output, Actual Output** |
| Part I | Guceanu George – Marian | *Domain Checker, Login* | **Domain Checker**: **Input**: Enter a domain name < **Expected Output**: Domain availability or error message **Actual Output**: Error page - not taking domain registrations anymore |
|  |  |  | **Login**: **Input**: Enter Username and Password **Expected Output**: Redirect to home page, user logged in **Actual Output**: Could not be tested because of register page issue |
| Part I | Forminte Andrei | *Register Page, Website Builder* | **Register Page**: **Input**: Enter Member name, Password, Email address **Expected Output**: Validation error or success message **Actual Output**: Redirect to payment page - design error |
|  |  |  | **Website Builder**: **Input**: Click on "Tour Our Website Builder" button **Expected Output**: Website Builder presentation **Actual Output**: Website Builder presentation |
| Part II | Guceanu George – Marian | All features | **All features**: **Input**: Perform exploratory testing on various features **Expected Output**: Functionality as per requirements, uncovering hidden issues **Actual Output**: Various issues found, such as empty advertisement section and empty template detail page |

# Test Report



# Issue Reporting

RIMGEA Strategy Application

**RIMGEA** stands for:

* **R**ecord
* **I**dentify
* **M**itigate
* **G**enerate
* **E**valuate
* **A**ct

**1. Record:**

* **Issue Description:** During testing, it was observed that the Register button redirects users to a payment page instead of providing a success message or validation errors.
* **Feature Affected:** Registration Page
* **Test Case:** Input: Member name, Password, Email address; Expected Output: Validation error or success message; Actual Output: Redirect to payment page

**2. Identify:**

* **Issue Type:** Design Issue
* **Priority:** High
* **Impact:** This issue prevents new users from successfully registering and accessing the application, which is a critical functionality.

**3. Mitigate:**

* **Temporary Solution:** Inform users via a temporary message on the registration page that there is an issue with registration and advise them to contact support for account creation.
* **Long-term Solution:** Redesign the registration workflow to ensure proper validation and success messaging.

# Conclusions. Lessons Learned

The testing project successfully identified several critical issues, particularly in the registration and login functionalities. These issues significantly impact the user experience and need immediate attention to ensure the application's usability and reliability.

The application of various test design techniques, including scenario-based testing for explicit and implicit specifications and session-based exploratory testing, provided comprehensive coverage. This approach helped uncover both evident bugs and hidden issues that could affect the application's performance.

The exploratory testing sessions proved invaluable in identifying unexpected issues, such as empty advertisement sections and design inconsistencies. This underscores the necessity of including exploratory testing in the test strategy to uncover defects that structured testing might miss.

The collaboration between team members, Guceanu George – Marian and Forminte Andrei, was crucial in covering all features effectively. The joint effort ensured thorough testing across different dimensions, from functional to usability aspects.

Lessons Learned:

1. **Early Detection of Design Flaws**: Early detection of design flaws, like the registration redirect issue, emphasizes the need for thorough initial requirement analysis and design reviews. Identifying such issues early can save significant time and resources in later development stages.
2. **Comprehensive Test Coverage**: Ensuring comprehensive test coverage through multiple techniques and strategies is essential. The combination of scenario-based and exploratory testing provided a holistic view of the application's functionality and user experience, revealing issues that single-method testing might overlook.
3. **User Experience Focus**: Focusing on the user experience is critical. The issues found during testing, such as the misleading registration process and inconsistent design elements, highlighted the importance of designing intuitive and user-friendly interfaces. Usability testing should be an integral part of the testing process.
4. **Continuous Improvement in Testing Processes**: The project highlighted areas for improvement in our testing processes. Documenting test cases meticulously, maintaining clear communication among team members, and regularly updating testing strategies based on findings can lead to more effective and efficient testing cycles.
5. **Need for Better Error Handling**: Proper error handling mechanisms need to be integrated throughout the application. The errors and unexpected redirects encountered during testing suggest the need for more robust validation and error messaging systems to guide users and provide clear feedback.
6. **Security and Performance Considerations**: The importance of security and performance testing was underscored by the need to ensure the website's security and acceptable loading times. Future projects should incorporate dedicated security and performance testing phases to address these critical aspects comprehensively.