MIDDLE EAST TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

DEĞİŞ TOKUŞ - ONLINE BARTER MARKET

SOFTWARE TEST DOCUMENT

LONESOME CODEBOYS

Ali Can BATUR	1745793
Emre DENİZ	1745876
Donny Irawan BULHADIE	1702240
Ismail Sarp DİKKAYA	1745884

Preface

This document includes test specifications, test types and test cases for Değiş Tokuş – Online barter market system and the document is prepared according to the "IEEE Standard for Software and System Test Documentation – IEEE Std 829 – 2008".

This document provides complete test cases for Değiş Tokuş – Online barter market system.

The first section of the document contains introduction and scope of the document which specifies the intended audience for the document.

The second section includes details for the system test plan, general features to be tested and not tested and testing approach taken when the test is conducted.

The third section includes test management, and the fourth section contains details of each of the test cases.

The final section shows the general test process, overall results and comments about test cases.

Table of Contents

List of Tables

1. Introduction

To describe all the testing that had been conducted for Değiş Tokuş web based application, this Software Test Document (STD) were prepared in order to ease the testers to design and perform the essential test that would help improving the quality of the application. Moreover, It can also be a proof that the system that we build is working as same as the requirement that mentioned in System Requirement Specification (SRS).

1.1. Document Identifier

This is the Version 1.0 of Software Test Document for Değiş Tokuş web based application.

Intended audience will be the evaluator of METU computer engineering graduation project and the users of the project.

IEEE Std 829-2008 is used as reference for this Software Test Document.

1.2. Scope

The detail description of the testing phases will be provided in this document. It will mainly test the objective of the program at the first place by observing their input and output of the test. The documentation will be limited to System Testing as the project is not in the big scale system beside System testing is sufficient for this kind of project.

The test mainly will be based on the System Requirement Specification. Therefore, the functional requirement and the user cases will be tested on the product whether it satisfies the requirement or not.

Inspection, Analysis, demonstration, verification, and validation will be considered while testing each test case.

1.3. References

- Değiş Tokuş, Software Requirements Specification (SRS) - 29.12.2013,
- Değiş Tokuş, Software Design Descriptions (SDD) - 29.12.2013,
- IEEE Std 829-2008

1.4. Level in the Overall Test

There are two kind of test that we conducted for this product, which are iteration test and full pass test. However, this System Test Document (STD) will only include the detail of the test of full pass test as it already includes the test that had been conducted in iteration test.

1.4.1. Iteration Test

- **Frequency:** These tests are conducted in each iteration progress that is already scheduled. Also, these tests are conducted for observing that the new feature that added into the product whether it works as expected or not.
- **Scope:** The test verifies the integration of new features and the regression for the existing ones. Manual test cases are used for these tests. Therefore, the tester will use the new feature by trying an unexpected behavior to the product.
- **Coverage:** The Iteration test cover the functionalities up to where the product has been developed with the new feature included.

1.4.2. Full pass Test – Pre-release

- **Frequency:** This test is done before the product announce as a finish product or usable. It is to make sure whether the product is working well as expected.
- **Scope**: Performance tests are run and also distribution validation tests. The test cases ran covers all the functionalities of the product.
- **Coverage**: Full past test covers entire system function to be test. Mostly, test is done manually but there was an automated test at some functions too (Selenium). And also the test is limited to system testing, as it is the most necessary one to be test for this kind of project.

1.5. Test Classes and Overall Test Conditions

Test classes are divided according to the functional requirements that can be found in SRS document of Değiş Tokuş.

Değiş Tokuş web application is tested on different machines capable of running different web browsers including Internet Explorer, Mozilla Firefox, Opera and Google Chrome.

2. Details for System Test Plan

2.1. Test Items and Their Identifiers

Değiş Tokuş is a web based application that can be accessed through Internet connection with browser. Thus, there will be no installation instruction available or to be installed in order to use the system. Furthermore, Değiş Tokuş is an independent system. Therefore, the only test item is the Değiş Tokuş System itself.

2.2. Test Traceability Matrix

Use cases Test Cases	Register UC	Login UC	Add Item UC	Remove Item UC	Search Item UC	Search User UC	Edit Profile UC	Message User UC
DegisTokus.TC.1 – 1, 2, 3, 4	x							
DegisTokus.TC.2 – 1, 2	x	x						
DegisTokus.TC.3 – 1, 2, 3	x	x	x					
DegisTokus.TC.4	x	x	x	x				
DegisTokus.TC.5 – 1, 2, 3	x	x			x			
DegisTokus.TC.6 – 1, 2, 3	x	x				x		
DegisTokus.TC.7 – 1, 2	x	x					x	
DegisTokus.TC.8 – 1, 2	x	x				x		x

Table 1: Test Traceability Matrix

2.3. Features to be Tested

These System test document will only test system function of the application. System Requirement Specifications document will be used to prepared the test cases to plot the test scenario. The system functions that will be test are:

- Register User
- Login User
- Add Item
- Remove Item
- Search Item
- Search User
- Edit Profile
- Message User

2.4. Features Not to be Tested

Interface Requirement and nonfunctional requirement will not be tested on this STD. However, it will still be inspected independently out of this document. The entire functional requirements that are mentioned in the SRS are tested and analyzed in this document.

2.5. Approach

We have used manual and automated tools (Selenium) for our test cases. These automated tools require no special training. Also, testing is done with different configurations. These configurations are listed as follows:

- Different web browsers (Google Chrome, Internet Explorer, Mozilla Firefox, Opera etc.)
- Different computers
- Same server (local) but different ports

2.6. Item Pass / Fail Criteria

The system that we tested is Değiş Tokuş System as our project. The criteria of pass and fail is decided after the entire test has been conducted. The system is decided as fail if one of the tests fails. Otherwise, it is a pass. Fail test case examined and fixed before the test start over again.

2.7. Test Deliverables

The items listed below can be considered as test deliverables:

- Software test document (STD).
- Test cases.
- Test design specifications.
- Tools and their outputs.
- Error logs and execution logs.
- Problem reports and corrective actions.

3. Test Management

3.1 Test Progression / Planned Activities and Tasks

Our testing mechanism is based on the features we added to our application. Basically, when we add a new feature to our system we tested that feature as a separate entity and then we have combined this feature with other feature of the system and tested them as a whole. That's why, we have followed a cumulative approach when we test our application, so the test progression is simply about testing each unit and then testing them as one entity.

When we saw an error after we test an entity, then we solved the problem before we integrate that feature to our system. Otherwise, we could face a situation such that we could lose our working features as well.

Also, if we did not see any problem after we tested that module separately but it became problematic after integrated with other modules, we fixed the problem and then we retested the module itself and with other modules as well.

Thus, as a result, the test progression is carried out as double testing, one for unit testing and one for system testing after implementation of each module of the system.

3.2 Environment

We used a PC as a platform that is capable of running different web browsers on itself.

4. Test case details

This section will explain the detail information about the test case for each functional requirement. Each test case includes objective, input, outcome, environmental needs, special procedural requirements and intercase dependencies.

4.1. Test Case: Register User

4.1.1 Test Case: Register User – 1

Test Case Id	DegisTokus.TC.1 - 1
Objective	This test case aims to verify whether all information that are entered on the form are saved on the database and can be used to log in to the system by the user afterward.
Input	User provides information that is required on the form on the sign up page.
Outcome	User Information will be saved on the database as a new account.
Environmental needs	Any internet browser with an internet connection.
Special procedural requirements	User should fill all-important information on the form that shown by the star sign.
Intercase Dependencies	-

Table 2: Test Case: Register User - 1

4.1.2. Test Case: Register User – 2

Test Case Identifier	DegisTokus.TC.1 - 2
Objective	This test case aims to verify whether error message will show up or not if one of the register form list is missed when a user fill the register form.
Input	User fill the register form but miss one or more form list on the form.
Outcome	Error message will come up and notify the User that they need to fill the entire form list that is required.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	-
Intercase Dependencies	-

Table 3: Test Case: Register User - 2

4.1.3. Test Case: Register User – 3

Test Case Identifier	**DegisTokus.TC.1 - 3**
Objective	This test case aims to verify whether error message will show up or not if the security verification question is answered wrongly.
Input	User fills all the form list that is required but wrongly answer the security verification question.
Outcome	Error message will come up and notify the user that they need to answer the security verification question correctly.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	Other form lists except for security verification question need to be filled correctly except for security verification question.
Intercase Dependencies	-

Table 4: Test Case: Register User - 3

4.1.4. Test Case: Register User - 4

Test Case Identifier	**DegisTokus.TC.1 - 4**
Objective	This test case aims to verify whether error message will show up or not if the register agreement is not checked.
Input	User fills the register form but do not check the register agreement.
Outcome	Error message will come up and notify the user that they need to check the register agreement.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	Other form lists need to be filled correctly.
Intercase Dependencies	-

Table 5: Test Case: Register User - 4

4.2. Test Case: Login User

4.2.1. Test Case: Login User - 1

Test Case Identifier	**DegisTokus.TC.2 - 1**
Objective	This test case aims to verify that user needs to provide valid account to login to the system.
Input	Username and user password that had been registered.
Outcome	System will confirm user account and login to the system.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	-
Intercase Dependencies	DegisTokus.TC.1.Register User should be done first before user can login. Except for those who already have an account beforehand.

Table 6: Test Case: Login User - 1

4.2.2. Test Case: Login User – 2

Test Case Identifier	**DegisTokus.TC.2 - 2**
Objective	This test case aims to verify that whether error message will come up if either the username or password wrongly typed.
Input	Username or user password that hasn't been registered.
Outcome	Error message will come up and notify the user that the username or password is wrong.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	-
Intercase Dependencies	DegisTokus.TC.1.Register User should be done first before user can login. Except for those who already have an account beforehand.

Table 7: Test Case: Login User - 2

4.3. Test Case: Add Item

4.3.1. Test Case: Add Item - 1

Test Case Identifier	**DegisTokus.TC.3 - 1**
Objective	This test case aims to verify whether all information that are entered on the form are saved on the database and can be searched or found by using search item menu afterward.
Input	User provides information that is required on the form on add item page.
Outcome	Item Information will be saved on the database.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start adding an item.
Intercase Dependencies	DegisTokus.TC.1.Register User and DegisTokus.TC.2.Login User should be done first before user can start adding an item.

Table 8: Test Case: Add Item - 1

4.3.2. Test Case: Add Item - 2

Test Case Identifier	**DegisTokus.TC.3 - 2**
Objective	This test case aims to verify the default category of an item when they are added without the user specification.
Input	User provides information that is required on the form on add item page except for the category option.
Outcome	Item Information will be save on the database with the default category.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start adding an item.
Intercase Dependencies	DegisTokus.TC.1.Register User and DegisTokus.TC.2.Login User should be done first before user can start adding an item.

Table 9: Test Case: Add Item - 2

4.3.3. Test Case: Add Item - 3

Test Case Identifier	**DegisTokus.TC.3 - 3**
Objective	This test case aims to verify that the default image will be added when user forget to add it.
Input	User provides information that is required on the form on add item page except for the image.
Outcome	Item Information will be save on the database with the default image as the picture of the item.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start adding an item.
Intercase Dependencies	DegisTokus.TC.1.Register User and DegisTokus.TC.2.Login User should be done first before user can start adding an item.

Table 10: Test Case: Add Item - 3

4.4. Test Case: Remove Item

Test Case Identifier	**DegisTokus.TC.4**
Objective	This test case aims to verify whether the selected item to be deleted is deleted both on the website and the database.
Input	Users select the item that that they want to delete.
Outcome	The item is deleted from the database and also on the website.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start deleting an item.
Intercase Dependencies	DegisTokus.TC.1.Register User (except for those who already have an account beforehand), DegisTokus.TC.2.Login User and DegisTokus.TC.3.Add Item should be done first before user can start deleting an item.

Table 11: Test Case: Remove Item

4.5. Test Case: Search Item

4.5.1. Test Case: Search Item - 1

Test Case Identifier	DegisTokus.TC.5– 1
Objective	This test case aims to verify whether the search item function work well or as expected.
Input	Item keyword, category, and the location.
Outcome	Item that was searched will show up as a list of items.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start searching an item
Intercase Dependencies	DegisTokus.TC.1.Register User (except for those who already have an account beforehand) and DegisTokus.TC.2.Login User should be done first before user can start searching an item.

Table 12: Test Case: Search Item - 1

4.5.2. Test Case: Search Item - 2

Test Case Identifier	DegisTokus.TC.5 – 2
Objective	This test case aims to verify whether the search item function work well even though one of the key search (item's keyword, category, and location) is not specified.
Input	Item's keyword or category or the location.
Outcome	Item that was searched will show up as a list of items that fits the search specification.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start searching an item
Intercase Dependencies	DegisTokus.TC.1.Register User (except for those who already have an account beforehand) and DegisTokus.TC.2.Login User should be done first before user can start searching an item.

Table 13: Test Case: Search Item - 2

4.5.3. Test Case: Search Item – 3

Test Case Identifier	**DegisTokus.TC.5 – 3**
Objective	This test case aims to verify whether the result of the searched item will be sorted alphabetically as a list.
Input	Item’s keyword or category or the location.
Outcome	Item that was searched will show up as a list of items that fits the search specification. And the list will be sorted alphabetically.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start searching an item
Intercase Dependencies	DegisTokus.TC.1.Register User (except for those who already have an account beforehand) and DegisTokus.TC.2.Login User should be done first before user can start searching an item.

Table 14: Test Case: Search Item - 3

4.6. Test Case: Search User

4.6.1. Test Case: Search User – 1

Test Case Identifier	**DegisTokus.TC.6 – 1**
Objective	This test case aims to verify that search user function work well as expected.
Input	The keyword of the username and location
Outcome	User with the username and the location specified will show up.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start searching a user.
Intercase Dependencies	DegisTokus.TC.1.Register User (except for those who already have an account beforehand) and DegisTokus.TC.2.Login User should be done first before user can start searching a user.

Table 15: Test Case: Search User - 1

4.6.2. Test Case: Search User – 2

Test Case Identifier	**DegisTokus.TC.6 – 2**
Objective	This test case aims to verify that search user function work well as expected even though the location of the user searched is not specified.
Input	The keyword of the username only.
Outcome	User with the username and other usernames include specified username as substring will show up.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start searching a user.
Intercase Dependencies	DegisTokus.TC.1.Register User (except for those who already have an account beforehand) and DegisTokus.TC.2.Login User should be done first before user can start searching a user.

Table 16: Test Case: Search User - 2

4.6.3. Test Case: Search User – 3

Test Case Identifier	**DegisTokus.TC.6 – 3**
Objective	This test case aims to verify that the search result of the user search function will be sorted.
Input	The keyword of the username only or with the location also specified.
Outcome	User with the username specified will show up and sorted alphabetically.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start searching a user.
Intercase Dependencies	DegisTokus.TC.1.Register User (except for those who already have an account beforehand) and DegisTokus.TC.2.Login User should be done first before user can start searching a user.

Table 17: Test Case: Search User - 3

4.7 Test Case: Edit Profile

4.7.1 Test Case: Edit Profile - 1

Test Case Identifier	**DegisTokus.TC.7 - 1**
Objective	This test case aims to verify whether users can change their profile after making an account and login. This test case will focus on the user who would like add an extra information on their profile.
Input	Users fill the information that they have not filled yet.
Outcome	User extra information is saved on the database and the profile changes accordingly.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start editing their profile.
Intercase Dependencies	DegisTokus.TC.1.Register User DegisTokus.TC.2.Login User should be done first before user can start adding extra information on their profile.

Table 18: Test Case: Edit Profile - 1

4.7.2 Test Case: Edit Profile – 2

Test Case Identifier	**DegisTokus.TC.7 - 2**
Objective	This test case aims to verify whether Users can change their profile after making an account and login. This test case will focus on the user who would like to erase information on their profile.
Input	-
Outcome	User extra information is erased on the database and the profile changes accordingly.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start editing their profile.
Intercase Dependencies	DegisTokus.TC.1.Register User DegisTokus.TC.2.Login User should be done first before user can start erasing extra information from their profile.

Table 19: Test Case: Edit Profile - 2

4.8 Test Case: Message User

4.8.1. Test Case: Message User - 1

Test Case Identifier	**DegisTokus.TC.8 – 1**
Objective	This test case aims to verify that the message that was sent through the message features of Değiş Tokuş was sent correctly to the user that was specified.
Input	Username of the user that you would like to send the message to, subject of the message, and the message itself need to be provided.
Outcome	The message sent to the user that was specified.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start sending a message.
Intercase Dependencies	DegisTokus.TC.1.Register User and DegisTokus.TC.2.Login User should be done first before user can start sending a message. Except for those who already have an account beforehand.

Table 20: Test Case: Message User - 1

4.8.2. Test Case: Message User - 2

Test Case Identifier	**DegisTokus.TC.8 – 2**
Objective	This test case aims to verify that error message will be shown when one of the form list (Username of the user that you would like to send the message to, subject of the message, and the message itself) is not specified.
Input	One or more form list is not specified.
Outcome	Error message will be shown to notify that user need to fill the missing form.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	User need to login before they can start sending a message.
Intercase Dependencies	DegisTokus.TC.1.Register User and DegisTokus.TC.2.Login User should be done first before user can start sending a message. Except for those who already have an account beforehand.

Table 21: Test Case: Message User – 2

4.9 Test Case: SQL Injection

Test Case Identifier	DegisTokus.TC.9
Objective	This test case aims to verify that the system is not affected with the SQL injection.
Input	Input for database attack
Outcome	The system is no affected SQL Injection attacks.
Environmental needs	Internet browser with an internet connection.
Special procedural requirements	Over user login and all the other input fields.
Intercase Dependencies	-

Table 22: Test Case: SQL Injection

5. System Test Report Details

5.1 Overview of Test Results

At the end of each test cases, it is observed the system works as expected after each test cases. Thus, it can be said that, the system passed all the test cases provided. However, it does not mean that the system is flawless, there may be some cases that the system behaves unexpectedly. If such cases is discovered, problem fixing actions will be taken and new test cases will be prepared accordingly.

5.2 Results of the Test Cases

DegisTokus.TC.1 – 1	**Passed**
DegisTokus.TC.1 – 2	**Passed**
DegisTokus.TC.1 – 3	**Passed**
DegisTokus.TC.1 – 4	**Passed**
DegisTokus.TC.2 – 1	**Passed**
DegisTokus.TC.2 – 2	**Passed**
DegisTokus.TC.3 – 1	**Passed**
DegisTokus.TC.3 – 2	**Passed**
DegisTokus.TC.3 – 3	**Passed**
DegisTokus.TC.4	**Passed**
DegisTokus.TC.5 – 1	**Passed**
DegisTokus.TC.5 – 2	**Passed**
DegisTokus.TC.5 – 3	**Passed**
DegisTokus.TC.6 – 1	**Passed**
DegisTokus.TC.6 – 2	**Passed**
DegisTokus.TC.6 – 3	**Passed**
DegisTokus.TC.7 – 1	**Passed**
DegisTokus.TC.7 – 2	**Passed**
DegisTokus.TC.8 – 1	**Passed**
DegisTokus.TC.8 – 2	**Passed**

Table 23: Test Case Results

5.3 Conclusions and Recommendations

As a result, our web application passed all the tests we have provided, but it is not certain that the system has no bugs or works expectedly in any environment. Also, when new features are added in the future, module tests and integration tests will be done to make sure new features work together with old ones in an expected manner.

5.4 Glossary for Acronyms

IEEE	Institute of Electrical and Electronics Engineers
Online Barter Market	Type of the web application developed
User	A person who uses the system
GUI	Graphical user interface
Stuff/Good/Item	People's possessions that are shared, given or exchanged on the system
Değiş Tokuş	Name of the application given by group members
SRS	Software Requirements Specification
SDD	Software Design Description
STD	Software Test Document
METU	Middle East Technical University
PC	Personal Computer
PHP	Hypertext Preprocessor

Table 24: Acronyms

5.5 Document Revision History

Document ID	Date	Version	Status	Author
Değiş Tokuş - STD 1.0	20.05.2014	1.0	Created	Lonesome Codeboys

Table 25: Document Revision History