



MIDDLE EAST TECHNICAL UNIVERSITY



DEPARTMENT OF COMPUTER ENGINEERING

CENG 491 - SENIOR PROJECT

TESTING SPECIFICATION



PREPARED BY

<i>Aycan Tekerek 1347988</i>	<i>aycantekerek@gmail.com</i>
<i>Baris Yanar 1348093</i>	<i>barisyanar@gmail.com</i>
<i>Seniz Yıldırım 1502814</i>	<i>senizyildirim@gmail.com</i>
<i>Wai Phyoe Maung 1368687</i>	<i>waiphyoe2003@gmail.com</i>

TABLE OF CONTENTS

1. INTRODUCTION.....	3
1.1. Goals and Objectives	3
1.2. Scope of Document	3
1.3. Statement of Testing Plan Scope	3
1.4. Testing Constraints	4
1.4.1. Time.....	4
1.4.2. Staff	4
1.4.3. Platform Independency	4
2. TESTING PLAN AND STRATEGIES	4
2.1. Unit Testing	4
2.1.1. Job Seeker Modules	5
2.1.2. Job Provider Modules	7
2.2. Integration Testing	9
2.3. Validation Testing	10
2.3.1. Requirements Validation	10
2.3.2. Design Validation	11
3. RECORD KEEPING AND LOGS	11
4. STAFFING	12
5. TOOLS AND ENVIRONMENTS	13
6. TESTING SCHEDULE.....	13

1 INTRODUCTION

1.1 GOALS AND OBJECTIVES

‘ISTEIS’ is a SOA based gateway portal with various modules for different types of users which have to be developed and implemented in a limited period of time tied to a strict schedule. Throughout this schedule, we have to implement and integrate each of our modules on a consistent and properly functioning system. Errors and bugs in software projects especially in the projects implemented in a limited time are inevitable. However, our main goal is to release an error-free and correctly functioning final product meeting our project’s design requirements. As a result of this, testing is one of the most important parts of our process. Testing is always a challenging process however it helps to deliver usable and stable systems. We believe that by following our testing plan and strategy, we will be able to detect and correct the bugs in our product and satisfy the requirements.

1.2 SCOPE OF DOCUMENT

The scope of this document is to describe the testing process of the project ‘ISTEIS’. Actually, for our project the testing process takes place throughout the implementation period. Every member of our team is responsible for developing error and bug free modules. Although, service oriented architecture gives the flexibility of loose coupling, we also need the developed modules to be working accordingly and in coordination. In fact, while developing our project up to now, we did testing for each module, so we will explain about the testing process that took place since the beginning of the project and we will also state the future plans about testing process on this document.

1.3 STATEMENT OF TESTING PLAN SCOPE

There are various testing techniques in software engineering so we tried to determine which of these techniques are applicable to our project and meet our needs in limited time and tight schedule. At the end, we decided unit testing, integration testing, validation testing and higher order tests to be our testing techniques throughout our project’s development process.

Unit testing: For Signup and Login Modules of all users on our system, CV, Job Search and Job Apply Modules of job seeker users and Job Offer and Employee Search Modules of job provider users.

Integration testing: We need to be more careful on this stage because our system has many modules and each module has many more services in it. And these services need to be work in coordination.

Validation testing: We need to validate if our project requirements and design needs are fully satisfied.

Performance testing: Testing of methods used to improve performance and quality of the project.

Stress testing: Testing the control of data and its flow in our system.

Alpha testing: Testing of our portal by someone other than the group members.

1.4 MAJOR CONSTRAINTS

1.4.1 TIME

We have to complete our project's implementation with all of testing, removing bugs and errors stages in a short time. Therefore, time is one of the main constraints for testing process of our system. In order to complete the project in time after a successful testing period, we have to follow our testing schedule.

1.4.2 STAFF

Our team consists of four people who are responsible for both development and testing of the whole system. Also, we are working in groups of twos in two different development parts of our project. Therefore, staff is also a major constraint for testing process of the system.

1.4.3 PLATFORM INDEPENDENCY

One of the major properties of service oriented architecture is that it is platform independent. So, from the beginning of the project we aimed our system to be platform independent, cause it is a web based system this constraint is more important. To see whether our system works on different platforms we need to test it on each of them.

2 TESTING PLAN AND STRATEGIES

2.1 UNIT TESTING

Our development process is going on two different layers, so we have to make sure on both sides every module is working correctly to have a stable combined system at the end. On every side of the development, we believe that we ensure modules coordination easily by proceeding in this bottom up manner. All the services and their modules are tested in a black

box manner, by observing the input that is provided to the service and the output or the effect produced as a result.

2.1.1 JOB SEEKER MODULES

In order to test the Job Seeker Modules we developed some test scenarios according to the normal process flow of the module as follows. These test scenarios cover all the possible flows. If any one of these scenarios does not terminate successfully, we will trace and debug our services and correct accordingly.

Scenario 1

For any job seeker user who is not already signed up on to system;

- 1) “Üye Ol” button on the main page is clicked.
- 2) On the newly opened web page necessary fields are filled.
- 3) “Kaydet” button is clicked.
- 4) Then user will be directed to main page to login to system.

Scenario 2

For any job seeker user who is already signed up on to system;

- 1) The username and password fields for job seeker user are filled.
- 2) “Giriş” button is clicked.
- 3) The job seeker main page comes to screen showing the membership information.

Scenario 3

For any job seeker user who is already signed up on to system;

- 1) “Üyelik Bilgileri Göster” button on the main tab is clicked.
- 2) The information given in the signup process is shown on the page.

Scenario 4

For any job seeker user who is already signed up on to system;

- 1) “Üyelik Bilgileri Güncelle” button on the main tab is clicked.
- 2) The information given in the signup process is shown on the page.
- 3) Any needed changes on the fields are done.
- 4) “Güncelle” button is clicked.
- 5) Updated information is shown on the page.

Scenario 5

For any job seeker user already entered the system;

- 1) “CV Oluştur” button on the main tab is clicked.
- 2) The empty CV form is appeared on the screen.
- 3) The needed fields are filled.

4) “Kaydet” button is clicked.

Scenario 6

For any job seeker user already filled his/her CV information on the system;

1) “CV Göster” button is clicked.

2) The user’s CV information entered before is shown on the screen.

Scenario 7

For any job seeker user already filled his/her CV information on the system;

1) “CV Güncelle” button is clicked.

2) The already given CV information is shown on the screen.

3) The needed changes are made.

4) “Güncelle” button is clicked.

5) Updated CV information is seen on the screen.

Scenario 8

For any job seeker user already entered the system;

1) “İş Ara” button on the main tab is clicked.

2) The search parameters on the combo boxes are selected.

3) Keyword field is filled.

4) “Ara” button is clicked.

5) The search results are shown on the bottom of these search fields. The search result list contains the company names, positions, job offer ids and the web page name from which the result came.

6) If any of the results from the list is double clicked, the user sees the details of the corresponding job offer.

7) On the screen which shows detailed job offer definitions, there is a button called “Başvur”.

8) If this “Başvur” button is clicked, the application to the seen job offer is made.

Scenario 9

For any job seeker user already entered the system;

1) “Başvuruları Göster” button on the main tab is clicked.

2) The list of job offers to which the user has applied is seen.

3) If any of these job offers is selected, then the user sees the detail of the job offer.

4) On this screen also there is a button called “Sil”, if this button is clicked, the user takes back his/her application to that job offer.

5) On returning from application deletion, the updated list is shown on the screen.

2.1.2 JOB PROVIDER MODULES

In order to test the Job Provider Modules we also developed some test scenarios according to the normal process flow of the module like we did for Job Seeker Modules. These test scenarios cover all the possible flows. If any one of these scenarios does not terminate successfully, we will trace and debug our services and correct accordingly.

Scenario 1

For any job provider user who is not already signed up on to system;

- 1) “Üye Firma Ol” button on the main page is clicked.
- 2) On the newly opened web page necessary fields are filled.
- 3) “Kaydet” button is clicked.
- 4) Then user will be directed to main page to login to system.

Scenario 2

For any job provider user who is already signed up on to system;

- 1) The username and password fields for job provider user are filled.
- 2) “Giriş” button is clicked.
- 3) The job provider main page comes to screen showing the job offer add steps and listing already added job offers.

Scenario 3

For any job provider user who is already logged in to system;

- 1) On the job provider main page, any of the already added job offers on the list is double clicked.
- 2) The detailed definition of the job offer is seen.
- 3) Below the definition there is “Sil” button.
- 4) “Sil” button is clicked.
- 5) The selected job offer is deleted and the updated list is shown below.

Scenario 4

For any job provider user who is already logged in to system;

- 1) On the job provider main page, any of the already added job offers on the list is double clicked.
- 2) The detailed definition of the job offer is seen.
- 3) Below the definition there is “Güncelle” button.
- 4) After the needed changes are done, “Güncelle” button is clicked.
- 5) The selected job offer is updated.

Scenario 5

For any job provider user who is already logged in to system and selected a job offer that is already added to the system by the company;

- 1) “Adayları Göster” button is clicked.
- 2) The list of applicants to the offer is shown on the page.
- 3) If any of the applicants is selected by double clicking, the CV information of the corresponding job seeker comes to screen.

Scenario 6

For any job provider user who is already logged in to system;

- 1) “Bilgileri Göster” button on the main tab is clicked.
- 2) The information given in the signup process is shown on the page.

Scenario 7

For any job provider user who is already logged in to system;

- 1) “Bilgileri Güncelle” button on the main tab is clicked.
- 2) The information given in the signup process is shown on the page.
- 3) Any needed changes on the fields are done.
- 4) “Güncelle” button is clicked.
- 5) Updated information is shown on the page.

Scenario 8

For any job provider user who is already logged in to system;

- 1) “İşçi Ara” button on the main tab is clicked.
- 2) The provided fields are selected from the combo box and the keyword field is filled.
- 3) “Ara” button is clicked.
- 4) The search results are shown below the button.
- 5) If any of the results is double clicked, the CV information of the corresponding job seeker user is shown on the page.

Scenario 9

For any job provider user who is already logged in to system;

- 1) “İş ilanı ekle” button on the main tab is clicked.
- 2) The provided fields are filled.
- 3) “Ekle” button is clicked.

2.2. INTEGRATION TESTING

As all the project members, we clearly in awareness of the fact that all the modules work correctly and in coordination is really important for our project. So, as we proceed by completing some services, we immediately combine them on corresponding module and start integration testing process. In this bottom up testing model besides ensuring that the component modules are correct, we believe we won't have to deal with many errors on the base structure while proceeding.

1	Login Services for all users	Sign Up Services for all users	Login services needs information to be in conformance with the information given in sign up process.
2	Show and Update Information Services for all users	Sign Up Services for all users	Show and Update Information Services needs to be gather information given in Sign Up process.
3	Show Information Services for all users	Update Information Services for all users	Show Information services needs to reflect changes coming from Update Information Services.
4	CV Show and Update Services for Job Seeker Users	CV Creation Service for Job Seeker Users	CV Show and Update Services for Job Seeker Users needs to be gather information given in CV Creation process.
5	CV Show Service for Job Seeker Users	CV Update Service for Job Seeker Users	CV Show service needs to reflect changes coming from CV Update Service.
6	Job Search Service for Job Seeker Users	Job Offer Module for Job Provider Users.	Job Search Service for Job Seeker Users shows also results coming from Job Offer Module for Job Provider Users.

7	Job Apply Service for Job Seeker Users	Job Offer Module for Job Provider Users.	Job Apply Service for Job Seeker Users needs information coming from Job Offer Module for Job Provider Users.
8	Show, Delete and Update Job Offer Services for Job Provider Users	Add Job Offer Services for Job Provider Users	Show, Delete and Update Job Offer Services for Job Provider Users need to gather information given in Add Job Offer Services for Job Provider Users.
9	Show Job Offer Service for Job Provider Users	Delete and Update Job Offer Services for Job Provider Users	Show Job Offer Service for Job Provider Users needs to reflect changes made on Delete and Update Job Offer Services for Job Provider Users.
10	Employee Search Service for Job Provider Users	CV Module for Job Seeker Users	Employee Search Service for Job Provider Users should gather information from CV Module for Job Seeker Users.

2.3 VALIDATION TESTING

Validation testing is a whole system's test that shows the conformance of the implementation with the requirement analysis and design. The requirements stated in requirements analysis and design reports form a basis for validation testing process. In this phase, we will use black box testing and we have divided this testing stage into two groups:

2.3.1 REQUIREMENTS VALIDATION

The tests for validation with respect to the requirements will be black box tests. While doing these tests we consult our requirement analysis report that we have generated according to the customers' expectations and requirements. In our report, we determined the requirements of the project roughly as follows:

The functional requirements;

- Sign up and login to system
- Update information
- Search information

And non-functional requirements;

- Reusability
- Reliability and consistency
- Interoperability and platform independence

During the requirements validation, we will be testing whether all of the above functionalities and properties are provided by 'ISTEIS'.

2.3.2 DESIGN VALIDATION

During the implementation of our project, we try to satisfy the design issues stated in our final design report. By this way, we can provide the conformance of design issues with the written document and implemented components. We observed that there are some changes to these components due to the implementation constraints while applying the design validation tests. We use white box testing method during these processes because of the fact that we want to observe all the implemented services and modules with the ones we stated in the final design report.

3 RECORD KEEPING AND LOGS

As mentioned before, testing is one of the most important processes during development of a project. So, some kind of record keeping mechanism should also be used for identifying bugs or state of the system at any instant. We develop our project on two different platforms based on java programming language. We named these modules as Main Modules, Integration Server Module and Application Server Module. In order to be able to have a common record logging for these platforms we use a document consisting of rows and columns, whose rows correspond to tests done on any developed part of any module and columns give information about the test results as follows:

Column 1; Main Module: In this column on which platform the test is done is recorded.

Column 2; Tested Module: In this column the name of the tested module is recorded for each test.

Column 3; Test Date: In this column the date at which the test done is recorded for each test.

Column 4; Problematic component: If the found bug is in integration server service development side than its user module and service name is recorded. If it is in application designer part, then its class is recorded.

Column 5; Description of the Problem: In this column, the found bug is described properly for each test.

Column 6; Fix Date: In this column, the date at which the found bug is fixed is recorded for each test.

4 STAFFING

The task distribution for testing process to the team members can be shown as follows:

Integration Server Side Testing	User Login Modules, User Information Update Modules, User Search Modules	Şeniz Yıldırım
	User Sign Up Modules, Job Seeker CV Module, Job Seeker Job Application Module, Job Provider Job Offer Modules	Aycan Tekerek
Application Designer Side Testing	User Sign Up and Login Modules, Information Update Modules	Barış Yanar
	Job Seeker CV Module, Job Seeker Job Application Module, Job Provider Job Offer Module, User Search Modules	Wai Phyo Maung

5 TOOLS AND ENVIRONMENTS

We use the following tools and environment for testing process of 'ISTEIS'.

WebMethods Integration Server: We use developer tool of this platform to implement our services and its trace and debug options to test our services.

Eclipse – Application Designer: For application designer side testing, we use eclipse to compile and debug the java codes of our services generated by integration server code generator tool. And also, application designer layout painter plug-in for eclipse to test user interfaces.

Internet Explorer and Mozilla Firefox: We use these environments to see how our interfaces look and work on.

6 TESTING SCHEDULE

Our testing schedule can be shown as follows:

Testing Tasks	Start Date	End Date
Unit testing of each module	20.5.2008	27.5.2008
Validation Tests	27.5.2008	30.5.2008
Performance Tests	27.5.2008	29.5.2008
Alpha Tests	30.5.2008	5.6.2008
Bug Tracing and Correction	2.6.2008	8.6.2008