

Cansu Hekim – 1630888
Bekir Doğru – 1560085
Zaman Safari – 1572254
Hamid Siddiqi - 1634427

Human Resources Management System

Test Specifications Report

1.INTRODUCTION	ERROR! BOOKMARK NOT DEFINED.
1.1. GOALS AND OBJECTIVES	2
1.2. STATEMENT OF SCOPE	3
1.3. MAJOR CONSTRAINT	4
1.4. DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	5
1.5. REFERENCES	5
2. TEST PLAN	6
2.1. SOFTWARE TO BE TESTED	6
2.2. TESTING STRATEGY.....	6
2.2.1. <i>Unit testing</i>	7
2.2.2. <i>Integration Testing</i>	7
2.2.3. <i>Validation Testing</i>	7
2.2.4. <i>High-order testing</i>	8
2.3. TEST METRICS	8
2.4. TESTING TOOLS AND ENVIRONMENTS.....	8
3. TEST PROCEDURE.....	9
3.1. UNIT TEST CASES	9
3.2. INTEGRATION TESTING	10
3.3. VALIDATION TESTING.....	12
3.4. HIGH-ORDER TESTING (A.K.A. SYSTEM TESTING).....	13
3.4.1 <i>Performance Test</i>	13
3.4.2 <i>Alpha Beta Tests</i>	13
4. TESTING RESOURCES AND STAFFING	13
5. TEST WORK PRODUCTS	14
6. TEST RECORD KEEPING AND TEST LOG	14
7. ORGANIZATION AND RESPONSIBILITIES.....	14
8. TEST SCHEDULE.....	15

1. Introduction

HR Management System project satisfies the needs of the Human Resources Department of a company to manage employees' personal data (citizen identity number, name, surname, birthdate, birthplace, educational information etc.), annual leaves, payroll, trainings, skills, performance evaluation and so on. Since HRMS, which is sponsored by Siemens EC, is a project having a huge database, huge number of classes mapping to each database table, and the codes of the ZK framework basically implementing the user interfaces. Thus, In order to develop a consistent, durable, reliable and well-integrated system which will be introduced to the end users soon, there has to be a testing procedure with a satisfying procedure and plan. For this purpose, this TSR document is prepared. This document is prepared as like, at first lines there is a smooth introduction and then, the test plan which is a systematic approach to testing this system, and out test procedure which is a formal specification of test cases to be applied to our target program modules, are presented. After a while, information about testing resources and staffing, test record keeping and test log are given. Before coming to end line of the document, organization and responsibilities are debated and explained with details and at the final page our test schedule for the whole test period is attached.

1.1. Goals and objectives

As software testing is a fundamental component of software quality assurance and represents a review of specification, design and coding, thus the importance of software testing and its impact on software cannot be underestimated. While testing, our team will consider both functional and non-functional features of the system. We will manage a breaking strategy in order to overcome this issue, we will break the system into small pieces, and then construct back. The reason behind this strategy is that firstly we will make sure that basic components are working properly on their own as they are expected and secondly

we will make sure that these properly working small components are also capable of working together correctly.

With the help of the testing platform we are aiming to have our project with the following quality criteria being satisfied.

- Being error-free
- High performance
- Logically correctness
- Compatibility between modules
- Easy-to-use
- Genuinely real-time

1.2. Statement of Scope

This document briefly describes the testing process of HRMS project. The main aim of this document is to present the testing plan as it is stated before. It covers descriptions of the methods used in the testing process of HRMS, plan of the testing phase, procedures to be followed and techniques of keeping records during testing. During the development of the project HRMS, each component of every module is supposed to be tested by the responsible developer(s) right after its implementation is finished. However, since many modules including many components work interactive in HRMS, further separate testing of modules and testing of the whole system is absolutely necessary.

To overcome this issue, we will be focusing on;

- What is going to be tested, both functional and non-functional
- What are the constraints while testing
- How to handle user-facing bugs

- How to find internal bugs
- How to be assured that if the system is working as expected
- How the testing relates to our schedule
- How team members are responsible for different tests

As we have planned we are going to test our system by including unit testing, integration testing, validation testing, and high-order testing or system testing. System testing will be alpha-beta testing. The details for this testing procedure can be found in the following sections.

1.3. Major Constraint

➤ Time

Time is a major constraint since we have a limited time period to complete our project. Thus, the duration period that can be reserved for testing process will clearly affect the result.

➤ Data

Since HRMS is composed of many modules running concurrently and passing data to each other during the run, minimizing the amount of data being transferred between modules is a goal in the sense of data constraint.

➤ Number of People in the Team

Testing process is also limited with the number of people in the team since concurrent processes speed up the whole testing process. The testing process will be as fast-moving as number of people in the team can serve.

➤ Hardware Device Capabilities

Our project is partially dependent on hardware devices. For example it is dependent on CPU capabilities since we are trying to implement a real-time system. During both functional and behavioral testing, since we use hardware



devices that we can afford, testing and surely the future of the project will be affected.

➤ **Test Results over Costs**

Test results over cost ratio are also another constraint in testing. If a test is not useful as it costs, it will be postponed and more useful tests will be regarded as more important. This manner will also affect the testing procedure.

1.4. Definitions, Acronyms and Abbreviations

- TSR : Test Specifications Report
- HRMS: Human Resource Management System
- Bug: The common term used to describe an error, flaw, mistake, failure, or fault in a computer program or system.
- DDR : Detailed Design Report
- IDR : Initial Design Report
- RDR : Revised Design Report
- Admin: Administrator
- HR : Human Resource
- Siemens EC : Siemens Enterprise Communication
- RAM : Random Access Memory
- TRAC : an open source, web-based project management and bug-tracking tool

1.5. References

- Test Specifications Template
- IEEE Standard for Software Test Documentation
- Detailed Design Report that we prepared earlier
- Pressman, Roger S. Software Engineering: A Practitioner's Approach, Sixth edition.

New York, NY: McGraw-Hill

➤ http://en.wikipedia.org/wiki/Software_testing

2. Test Plan

This section describes the overall testing strategy that we are going to use for testing the HRMS project. As a team we are agreed to use two strategies for testing our project which are mentioned in the bottom lines. First, if a module or section is developed individually due to division by work, responsibility of testing components is up to developer. On the other hand, in system testing these parts are checked by all team members. Second, if team collaborates to implementation testing process will also be done collaborately.

2.1. Software to Be Tested

Our HRMS project is being developed for Siemens who wants to control and manage their employee's data in a more appropriate and neat way. With the help of the HRMS they will be able to manage their personal data, control mechanism to authorize and authenticate for the employees' entry in a better way. They will be able save all the data regarding to their employee's and users and will be able to edit these data anytime the wish. Meanwhile, they will be able to add/delete new user/employee to/from the system. Any record of the data they add, edit and delete will be kept in the HRMS database in a secure manner. All components and their integration and finally all system will be tested in testing phase.

2.2. Testing strategy

In this document, whole testing process of HRMS is described. We will a bottom-up approach test methodology. First the units are tested separately, and then their integrations are tested. After those application based tests, their validation tests from previous reports are declared. Finally, high-order tests are

explained. All steps of testing process, which are completed, still in progress, and planned to be made in the future will be stated in the following sections.

2.2.1. Unit testing

HRMS project includes two main components to undergo unit testing. First one is Database Handler and the other is Interface handler. Those two components are tested separately in order to make sure that each component does not have any internal errors and works properly.

2.2.2. Integration Testing

After all the modules pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. This specification is vital since although each module might be working individually, different implementations coming from different modules may conflict while running together which is to be located during integration testing and to be recorded for fixing.

Integration tests of our HRMS project include the integration of main units, namely Database and Interface Handler.

2.2.3. Validation Testing

Result of this test is going to show whether expectations in design is met or not. Our Software Requirements Specification document defines the functional and non-functional requirements of the HRMS Project. Every single requirement in this document must be considered one-by-one and the software must be guaranteed to satisfy all these requirements. Test cases and scenarios are defined and will be run to ensure the correct working of the project as a whole. The order of validation is the same as integration.

2.2.4. High-order testing

While computing unit testing and integration testing, we will also be computing some high-order testing, namely performance testing and alpha-beta testing.

2.2.4.1 Performance Tests

Performance Tests will be made to see whether the program runs in real time as it should be.

2.2.4.2 Alpha and Beta Tests

To perform Alpha testing we will assemble the four existing role of the HRMS project and make our accounts accordingly and will perform alpha test on role type. After this we will do the beta testing by the help of the employees of the Siemens EC to see whether the application has bugs or not.

2.3. Test Metrics

We will use the following metrics:

- Number Of Test Cases Executed
- Number Of Bugs Detected
- Number Of Bugs Fixed
- Number Of Priority Bugs Fixed

2.4. Testing Tools and Environments

Testing tools are important for the success of testing phase and naturally the success of product. In our unit testing phase Eclipse and HeidiSQL are used. Following paragraphs have detailed information about testing tools.

➤ **Eclipse debugger**

Eclipse debugger features are used for every module development. By tracing and correcting bugs and errors our system's stability is increased. Since our codes are written in eclipse environment it is easy to keep tracking any mistakes especially in unit testing phase.

➤ **HeidiSQL**

HeidiSQL is an open source tool used to handle the administration of the MySQL database. It allows users to view, modify or delete tables and their records in database. This tool will be used in order to check the correctness of the database and database related parts of the project.

As our HRMS project is an internet based project thus hardware won't be an issue for components of our model since network components are standard on most of the computers so the only test environment for HRMS application would be the operating system. As HRMS Application is supposed to work under both Windows and Linux through the existing internet browsers. In our tests, it will be guaranteed that all features of our application works under both 32-bit and 64-bit operating systems on any internet browsers without any problem.

3. Test Procedure

In this section, the test strategies and methods will be explained in detail. The process of using them and applying them to our project will be analyzed.

3.1. Unit test cases

These test cases constitute the very basic test cases that we are going to use in order to find the main beginning errors. Since our project mainly queries the database and display information via graphical user interface, unit test cases can be classified in two main groups namely Database Handler and Interface Handler. These are described in detail below.

Database Handler:

Our HRMS system uses a huge database containing database tables needed to keep information. There are three basic functions implemented in our project

namely add new information, edit the existing information and delete information. Database handler will check these three functions whether they are appropriately doing their work on the database in the intended way. Therefore our database handler test cases will be as follows.

- Whether each user can add new entry to the database successfully.
- Whether each user can edit the existing entries successfully.
- Whether each user can delete information successfully.

Interface Handler:

In order to manage the data, our project has a graphical user interface, designed in a user-friendly manner. Interface handler will answer the questions below:

- Whether the fields, buttons, grids etc. are in the right place on the page.
- Whether the buttons are doing their work in an intended way.

3.2. Integration testing

From the time that we integrate all the modules, the testing of the functionalities of the each role type comes on to the stage. Since our HRMS project has security issues both in and out of the scope of the employee of the Siemens EC, we have the specific permissions related to each user type (authorization) and authentication mechanism. Our Integration Testing Procedure is given below.

- Firstly, we will create four users who have the four role types namely, HR, manager, employee, admin.
- Then we will log in with the user names and passwords of each user type.

This tests whether the authentication mechanism works correctly.

- Then we will also try some wrong user name and/or wrong password. We expect an error message by trying this case.

After testing authentication mechanism we are going to test the capabilities of each role type:

Employee Role Type Test Cases:

- Can employee see his/her personal information successfully?
- Can employee edit his/her personal information successfully?
- Can employee evaluate personal information and his/her manager's performance successfully?

HR Role Type Test Cases:

- Can HR see all employees' personal data successfully?
- Can HR see all employees' performance evaluation successfully?
- Can HR edit all employees' personal information?
- Can HR add a new employee by entering required personal information of that employee successfully?

Manager Role Type Test Cases:

- Can manager see the list of all employees that work under his responsibility successfully?
- Can manager see the personal data of the employees' whom work under his/her responsibility?
- Can manager evaluate the performance of his/her employees' whom work



under his/her responsibility successfully?

Admin Role Type Test Cases:

- Can admin add a new user to the system successfully?
- Can admin introduce a new role type to the system successfully?
- Can admin edit the specifications of an existing role?
- Can admin add/remove a role type to/from any user successfully? To be more specific, if a user can be both an employee, in addition to that admin is capable of adding a new role type of manager to that employee. Also a user can be both manager and employee and admin is capable of deleting the role of employee from that user.

Finally, we will test all these functionality mentioned above for users who have multiple role types. Therefore we will test many combinations of the role types and the test cases described above.

3.3. Validation testing

Our HRMS is implemented according to the specifications explained in the documents of SRS, IDR, DDR, and RDR, which constitute the requirements of the company of Siemens EC. These specifications are negotiated with both Siemens EC and the academic staff of the course CENG492. Therefore, it is important for us to check whether the functions and capabilities that we mentioned in those documents are implemented successfully. As a result, we are going to use requirement traceability that we mentioned in RDR to check whether our implementation of HRMS provide the requirements of the company.

The test case functions to be checked here:

- Is the data design consistent with the reports?

- Is the architecture design consistent with the reports?
- Does the interaction between modules work correctly?

3.4. High-order testing (a.k.a. System Testing)

3.4.1 Performance Test

Since our HRMS uses a huge database manipulation and requires a listing of entries in some cases, we should test the performance of the system and RAM usage of the system. Therefore the test cases below should be held:

- Does the project work in real time?
- How fast can the queried data be achieved from database?
- What is the maximum number of the online users at the same time?

3.4.2 Alpha Beta Tests

Since we have four role types and four group members each of us took one role and have account on HRMS. By means of that, we are/will be doing the tests of each role easily (Alpha Testing).

On the other hand, we are planning to test the system with the help of employee of the Siemens EC (Beta Testing).

With the help of the alpha and beta testing we will clearly see the bugs of the system.

4. Testing Resources and Staffing

First, each group member implements his/her own part using the same environment defined in the specifications report then these code pieces are combined together and embedded to the main source code of the project. While combination process, each group member takes part and this combination process is held all together in order not to face with problems that can be occur due to the

changes on these combined implementations. Finally, development team members will hold system tests. Then, end-users take part in testing operations. The updates on the features and the functionalities of the system will be described in detail to the end-users.

5. Test Work Products

Test work product is mainly TRAC. Whenever one of the group members finds a bug on the system, he/she will immediately inform the other users via TRAC system. Thus each group member always checks the system on daily basis.

6. Test Record Keeping and Test Log

Testing procedures usually results with bugs. When a team member finds a bug, this bug will be recorded in TRAC. Also, progress of the bugs will be recorded in TRAC and when it is resolved, related ticket in TRAC will be completed. Also, during the improvement of the project, after fixing errors, our code will be saved to SVN, so, the application will be versioned and archived.

7. Organization and Responsibilities

Our HRMS has four main modules namely, A&A, APPL, PDM and PEM. All the test methods and test strategies described in previous sections will be used in the scope of the testing phase of each module by each of the four group members.

- Bekir Dođru will test the A&A module.
- Zaman Safari will test the APPL module.
- Hamit Siddiqi will test the PDM module.
- Cansu Hekim will test the PEM module.

8. Test Schedule

A detailed test procedure schedule can be found on the following table:

Task	Dates
Unit Testing	May 7-May 12
Integration Testing	May12 - May 21
Validation Testing	May21-May 28
System Testing	May 28- June 7

Table 1: Test Schedule