**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Job Searching System**

|  |  |
| --- | --- |
| **Group JSS** | |
| **Group members** | Huỳnh Đức Tài – Leader – SE60630  Nguyễn Ngọc Thiện – Member – SE90129  Phạm Quốc Thái – Member – SE60599  Võ Hoàn Thiện – Member – SE60788  Trịnh Văn Hoàng – Member – 01769  Trần Chung Thành – Member – 01611 |
| **Supervisor** | Mr. Trần Đình Trí |
| **Ext. Supervisor** | N/A |
| **Project code** | JSS |

-Hanoi, 07/2015-

Table of contents

Contents

[Signature page 2](#_Toc425365526)

[Definitions, Acronyms, and Abbreviations 4](#_Toc425365527)

[5. System implementation & Testing 4](#_Toc425365528)

[5.1. Introduction 4](#_Toc425365529)

[5.1.1. Overview 4](#_Toc425365530)

[5.1.2. Test Approach 4](#_Toc425365531)

[5.2. Database Relationship Diagrams 5](#_Toc425365532)

[5.2.1. Physical Diagram 5](#_Toc425365533)

[5.2.2. Data Dictionary 6](#_Toc425365534)

[5.3. Performance measures 7](#_Toc425365535)

[5.4. Test Overview 7](#_Toc425365536)

[5.4.1. Test Approach 7](#_Toc425365537)

[Unit testing 8](#_Toc425365538)

[Integration testing 8](#_Toc425365539)

[System testing 8](#_Toc425365540)

[Acceptance testing 8](#_Toc425365541)

[5.5. Test Plan 8](#_Toc425365542)

[5.5.1. Features to be tested 8](#_Toc425365543)

[5.5.2. Features not to be tested 9](#_Toc425365544)

[5.5.3. Testing Tool and Environment 9](#_Toc425365545)

[5.6. Test Case 10](#_Toc425365546)

[5.6.1. Register 10](#_Toc425365547)

[5.6.2. Login 10](#_Toc425365548)

[5.6.3. …. 10](#_Toc425365549)

[5.7. Checklists 11](#_Toc425365550)

[5.7.1. Checklist of Validation 11](#_Toc425365551)

[5.7.2. Checklist of Submission 11](#_Toc425365552)

[5.7.3. System Test Check list 11](#_Toc425365553)

# Signature page

**Author:** Phạm Quốc Thái Date: 2015/07/15

*Test Leader*

**Reviewers:** Huỳnh Đức Tài Date:

*Project manager*

**Approval:** Trần Đình Trí Date:

*Supervisor*

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| JSS | Job Searching System |
| SRS | Software Requirement Specification |

# System implementation & Testing

## Introduction

### Overview

This section provides in detail all necessary information about implementation and testing procedure of JSS includes test plan, test cases and test result.

### Test Approach

* Goal: to test the whole system based on the SRS document.
* Method: system testing, black-box testing.

## Database Relationship Diagrams

### Physical Diagram

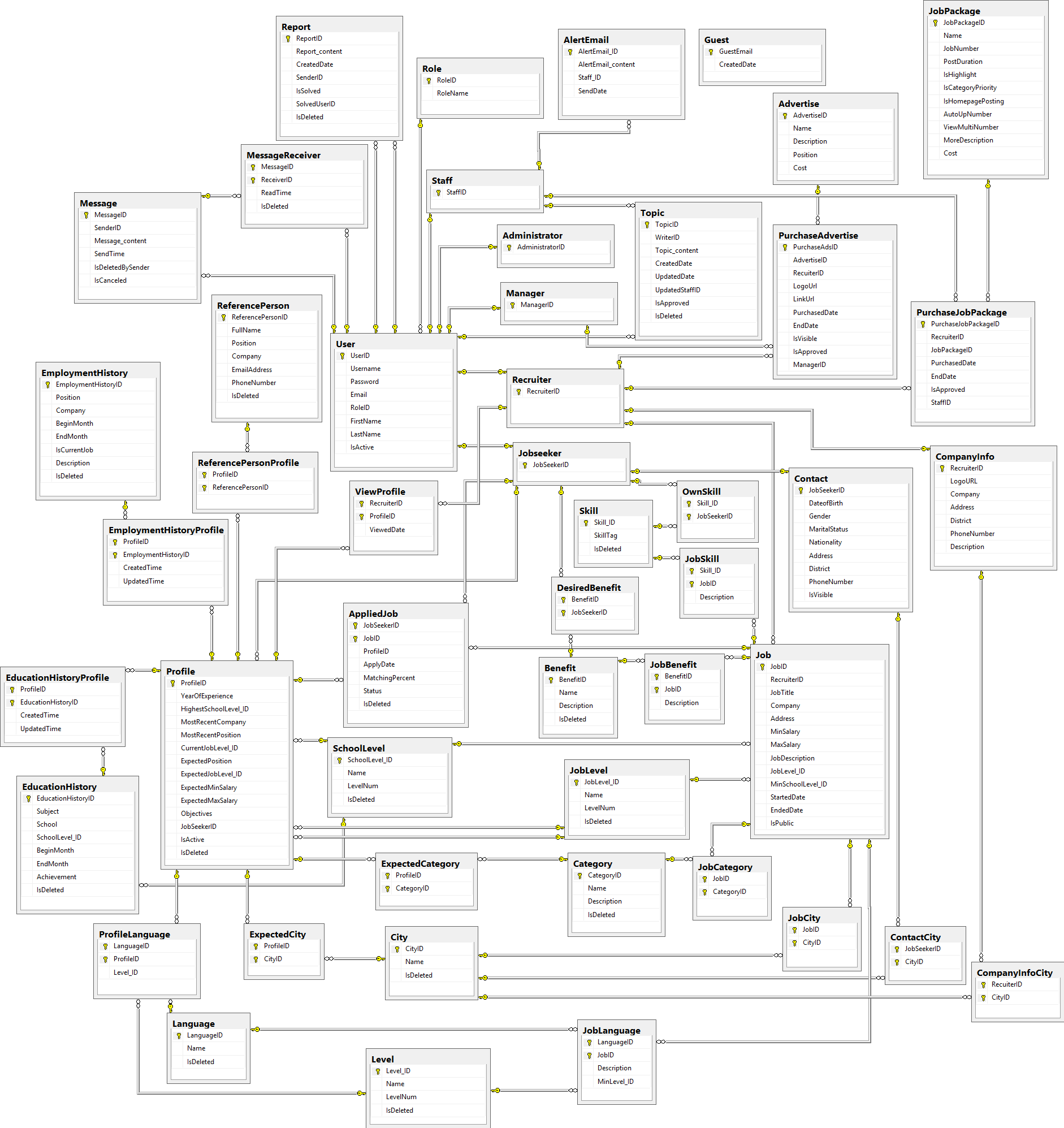


Figure 1: Physical Database Diagram

### Data Dictionary

|  |  |
| --- | --- |
| **Data dictionary: describe content of all tables** | |
| **Entity Name** | **Description** |
| User | Describe all users in the system. |
| Role | Describe all role in the system. |
| Jobseeker | List of users whose role are jobseeker. |
| Recruiter | List of users whose role are recruiter. |
| Staff | List of users whose role are staff. |
| Manager | List of users whose role are manager. |
| Administrator | List of users whose role are administrator. |
| Message | Store all the message in the system. |
| MessageReceiver | Mark the message with are received by user. |
| Report | Store all reports. |
| Topic | Store all written topic. |
| Contact | Contain contact information of jobseeker. |
| ContactCity | Map contact information with the city on the city list. |
| City | Contain list of avaiable cities. |
| JobCity | Map the job with the city on the city list. |
| ExpectedCity | The city which user want to work at. |
| CompanyInfoCity | Map the company info with the city list. |
| OwnSkill | Map user with some skills on the skill list. |
| Skill | Contain list of avaiable skills. |
| JobSkill | Map job with some required skills on the skill list. |
| DesireBenefit | Map user expected benefit with the benefit list. |
| Benefit | Contain list of benefit. |
| JobBenefit | Map the job providing benefit with the benefit list. |
| AppliedJob | Store information when user apply for a job. |
| Profile | Describe job seeker CV. |
| ViewProfile | Recruiter view CV of jobseeker. |
| ReferencePersonProfile | Map jobseeker CV with reference person information. |
| ReferencePerson | Conatin detail information of jobseeker. |
| EmploymentHistoryProfile | Map jobseeker CV with Employment. |
| EmploymentHistory | Contain information of employment history of jobseeker. |
| EducationHistoryProfile | Map jobseeker CV with education information. |
| EdutcationHistory | Contain information of education history of jobseeker. |
| SchoolLevel | Describe level of certification of jobseeker. |
| ProfileLanguage | Map the language of jobseeker on profile with the language list. |
| Language | List of avaiable language. |
| Level | List of available level of language. |
| JobLanguage | Map the required language of job with the language list. |
| Job | Describe all job information in the system. |
| JobCategory | Map the job with the job category. |
| Category | List all availble category of job. |
| ExpectedCategory | Map the expected working job category with the job category. |
| CompanyInfo | Store the information of all the company in the system. |
| PurchaseAdvertise | Store transaction when recruiter request to buy advertise. |
| Advertise | Describe all advertise in the system. |
| PurchaseJobPackage | Store transaction when recruiter request to buy job package. |
| JobPackage | Describe all job package in the system. |
| AlertEmail | Staff send job alert email. |
| Guest | Store email address of guest after they using Get Job Alert function. |

## Performance measures

N/A

## Test Overview

### Test Approach

V-model be used to implement testing process.



Figure ‑ V-model

Because requirement of project must be always suitable with user, V-model is implemented to testing process. In V-model, all process is divided into two groups corresponding to different stages: development and testing. Each stage of development will be combined with a corresponding test phase. The testing activities are focused and implemented in parallel with the activities related to requirements specification and design. In other words, this model encourages the activities related to the test plan was conducted early in the development cycle, not wait until the end stage reality. Testing progress is divided to 4 test level include: Unit test, Integration test, System test and Acceptance test.

### Unit testing

* In unit testing checks isolate each part of the program and ensure that part run as required. Unit tests help ensure the accuracy of the program, it helps establish constraints and parts of code we have done exactly that constraints.
* Unit test will be done by the developers.

### Integration testing

* In integration testing checks whether all the modules are combined or not combined together do the work have achieved results document requirements have been identified.
* Integration testing will be done by tester.

### System testing

* Compare the system specifications against the actual system.
* System testing checks if the integrated product meets the specified requirements.

### Acceptance testing

* In this test, the software will be done from the user check to find out if the software matching the expectations of users and perform as expected.

## Test Plan

The purpose of this section is to verify and ensure that JSS meets its design specification and other requirements from user. The following part will describe which features to be tested and which will not.

### Features to be tested

We will test the entire system based on SRS document.

(Reference: **Table 3-1: Functional Requirement**).

Base on the list of functions, all following features will be tested, which can include one or more functions. These features will be focused and tested thoroughly during the test phase.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function to be tested | Guest | JobSeeker | Recruiter | Staff | Manager | Administrator |
| Register | ✓ | - | - | - | - | - |
| Login | - | ✓ | ✓ | ✓ | ✓ | ✓ |
| Logout | - | ✓ | ✓ | ✓ | ✓ | ✓ |
| Get Job Alert | ✓ | ✓ | - | - | - | - |
| Find Job | ✓ | ✓ | - | - | - | - |
| View Job Detail | ✓ | ✓ | - | - | - | - |
| View Company Detail | ✓ | ✓ | - | - | - | - |
| Save Job | - | ✓ | - | - | - | - |
| Apply Job | - | ✓ | - | - | - | - |
| View/Delete Applied Job | - | ✓ | - | - | - | - |
| Manage Own CV | - | ✓ | - | - | - | - |
| Manage Own Skill | - | ✓ | - | - | - | - |
| Manage Company | - | - | ✓ | - | - | - |
| View Job Seeker’s CV | - | - | ✓ | - | - | - |
| Buy Job Package | - | - | ✓ | - | - | - |
| Manage Job | - | - | ✓ | - | - | - |
| Buy Advertise | - | - | ✓ | - | - | - |
| View Transaction | - | - | ✓ | - | - | - |
| Manage Applicant | - | - | ✓ | - | - | - |
| Process Job Package Request | - | - | - | ✓ | - | - |
| Manage Common List | - | - | - | ✓ | - | - |
| View Statistic Report | - | - | - | - | ✓ | - |
| Manage Advertise | - | - | - | - | - | ✓ |
| Manage Advertise Cost | - | - | - | - | ✓ | - |
| Process Advertise Request | - | - | - | - | ✓ | - |
| Manage Job Package | - | - | - | - | ✓ | - |
| Manage Account | - | - | - | - | - | ✓ |
| Send Report | - | ✓ | ✓ | - | - | - |
| Manage Report | - | - | - | ✓ | ✓ | ✓ |
| Manage Own Message | - | ✓ | ✓ | ✓ | ✓ | ✓ |
| Write Topic | - | ✓ | ✓ | ✓ | ✓ | ✓ |
| Manage Topic | - | - | - | ✓ | - | - |

### Features not to be tested

Taking advance from the number of members, our team will spend all effort to test every single functions of the system. There will be no feature not to be tested.

### Testing Tool and Environment

In order to process test case, test team must install all of the following software:

* Web browser: Google Chrome (at least version 39.0.2171.71)   
  or Firefox (at least version 33.1.0.5423) and must support JavaScript and HTML 5.
* Visual Studio 2013 and Microsoft SQL Server 2013 for running Source code.
* Tortoise SVN for management testing related document.

## Staff Test Case

### Create city

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Inter-test Case Dependence** | **Result** | **Test Date** | **Note** |
| CC01 | Create new city with blank "Tên thành phố" | 1- On the Create page, do not input "Tên thành phố"  2- Click "Tạo mới" | - After step 2, message “Thông tin này bắt buộc" is show under "Tên thành phố". |  | Passed | 22/7/2015 |  |
| CC02 | Create new city with content greater than 50 characters at "Tên thành phố " | 1- On the Create page, input content greater than 50 characters at "Tên thành phố "  2- Click "Tạo mới" | - After step 2, message “Nội dung nhập vào không vượt quá 50 kí tự" is show under "Tên thành phố". |  | Passed | 22/7/2015 |  |

### Update city

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Inter-test Case Dependence** | **Result** | **Test Date** | **Note** |
| UC01 | Update new city with blank "Tên thành phố" | 1- On the Update page, do not input "Tên thành phố"  2- Click "Cập nhật" | - After step 2, message “Thông tin này bắt buộc" is show under input "Tên thành phố". |  | Passed | 22/7/2015 |  |
| UC02 | Update new city with content greater than 50 characters at "Tên thành phố " | 1- On the Update page, input content greater than 50 characters at "Tên thành phố "  2- Click "Tạo mới" | - After step 2, message “Nội dung nhập vào không vượt quá 50 kí tự" is show under "Tên thành phố". |  | Passed | 22/7/2015 |  |

### ….

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Inter-test Case Dependence** | **Result** | **Test Date** | **Note** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

## Checklists

### Checklist of Validation

### Checklist of Submission

### System Test Check list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Question | Yes | No | N/A | Note | Priority | Severity |
| Is the system test using the unit tests and incremental integration test cases, requirements and design? |  |  |  |  |  | 2 |
| Have volume tests been included? |  |  |  |  |  | 3 |
| Have concurrency tests been included? (Multi user tests) |  |  |  |  |  | 3 |
| Have error-logging conditions been tested? |  |  |  |  | Mandatory | 2 |
| Have restart features been tested? |  |  |  |  | Mandatory | 2 |
| Have invalid input conditions been tested? |  |  |  |  | Mandatory | 1 |
| Have invalid output conditions been tested? |  |  |  |  | Mandatory | 1 |
| Has the functionality as per requirements been tested? |  |  |  |  | Mandatory | 1 |
| Have performance requirements been tested? |  |  |  |  | Mandatory | 1 |
| Have security requirements been tested? |  |  |  |  |  | 1 |
| Have the test results regarding security requirements been reviewed? |  |  |  |  |  | 3 |
| Have fail over conditions been tested? |  |  |  |  | Mandatory | 2 |
| Have scalability tests been conducted? |  |  |  |  |  | 4 |
| Is the correct version of the requirements, design, test case being used for system test ? |  |  |  |  |  | 1 |