

**FACULTY OF COMPUTING AND INFORMATION MANAGEMENT**

**BACHELOR OF INFORMATION SECURITY AND FORENSICS**

**UNIT: BISF 2208**

**PROJECT TITLE: ONLINE IQ TEST MANAGEMENT SYSTEM**

**SOFTWARE DESIGN SPECIFICATIONS DOCUMENT**

**BY: MUNENE JOHN**

**REG NO. 20/04112**

**SUPERVISOR: CATHERINE GACHIRI**

**Table of Contents**

[1.0 Introduction 3](#_Toc2061903619)

[1.1 Goals and objectives 3](#_Toc50422822)

[1.2 Statement of scope 3](#_Toc573431004)

[1.3 Software context 3](#_Toc1431446740)

[1.4 Major constraints 4](#_Toc1002705880)

[2.0 Data design 4](#_Toc1731189006)

[2.1 Global data structure 4](#_Toc1391822799)

[2.2 Temporary data structure 4](#_Toc930784137)

[2.3 Database description 4](#_Toc43893045)

[3.0 Architectural and component-level design 4](#_Toc517065889)

[3.1 System Structure 4](#_Toc1998485019)

[3.1.1 Architecture diagram 4](#_Toc1490465570)

[3.2 Description of components 5](#_Toc1709423968)

[3.2.1 Description of login screen 5](#_Toc1974337963)

[3.2.2 Description of New User Account Screen 6](#_Toc1679839447)

[3.2.1 Description of main menu 7](#_Toc1097380567)

[3.3 Database Modelling 7](#_Toc1289056734)

[4.0 User interface design 8](#_Toc1597035707)

[4.0.1 Expected Input: 8](#_Toc583940080)

[4.0.2 Output 8](#_Toc918544876)

[4.1 Objects and actions 9](#_Toc1749844036)

[4.1.1 Applicants use Case 9](#_Toc1500254699)

[4.1.2 Company Use Case 9](#_Toc2031273873)

[4.1.3 Admin use case 10](#_Toc678123872)

[5.0 Restrictions, limitations, and constraints 10](#_Toc88478301)

[6.0 Testing Issues 10](#_Toc1338046709)

[6.1 Classes of tests 10](#_Toc1823065723)

[6.2 Expected software response 11](#_Toc428512745)

[6.3 Performance bounds 11](#_Toc1775479286)

[6.4 Identification of critical components 11](#_Toc1379970982)

[References 11](#_Toc1297190787)

1.0 Introduction

This is the Software Design Specification (SDS) for the Online IQ Test Management System. The SDS will break down the project into components to describe in detail what the purpose of each component is and how it will be implemented.This design will detail the implementation of the requirements as defined in the Software Requirements Specification provided earlier. The SDS will also serve as a tool for verification and validation of the final product.

## 1.1 Goals and objectives

The Online IQ Test Management System is established to eliminate overcrowding and the spread of Covid-19. Its purpose is to establish an environment where there will be ease in conducting of interviews and keeping organised data in a database. This system will record number of applicants who attempted the IQ test and also save their scores in the database. It also notify’s the applicant if they qualify for the next step of interview.

## 1.2 Statement of scope

This Design Specification is to be used by Software Engineering and Software Quality Engineering as a defination of the design to be used to implement the Online IQ Test Management System.

The scope of the Online IQ Management System includes its distinct features, its benefits, and its limitations. The system's distinct features allow it to testing one’s IQ by processing attained through scores towards that particular field. The system enables the user to determine if they reach set score for the qualification of next step of the interview. This system extends the functionality of the old way of conducting interviews that is currently active. Additional fields and features will be added; Transforming the whole system to be an online system. The system provides a test that will be available for any applicant but the ones who will pass the test will have a chance to go to the next step of the interview which is onsite interview.

## 1.3 Software context

**Template** - Some modified template from BootStrapSite.

**Process** - One instance of a workflow.

**Task** - One step or piece of a workflow.

**Test Score Reporting** - Displays the attained of the applicant and records that in the database.

1.4 Major constraints

This application can only run on a system that supports Python3 with Django Framework must be installed. But hopefully to host the site on PythonEveryWhere.org on a unix web server. So that it can be available to all platforms including the mobile devices.

2.0 Data design

The design decision to break the system up into a three-module system was made to promote modularity within separate parts of the system. Modularity provides encapsulation for the important pieces of the system. Using encapsulation, we are able to change important parts of the system without changing the whole system.

2.1 Global data structure

If company A wants to use this system they will have to provide the company info and also licenses. This data will be stored in the database for future reference. If the company fails to provide that info then they will not be able to use the system.

2.2 Temporary data structure

The temporary data identified is the scores of the failed tests. This data will be temporary and not stored for a long time inorder to avoid system redanduncy.

## 2.3 Database description

This system will use an SQL database. The database will store; applicant’s details, the test, company’s details, system manual and test instructions.

3.0 Architectural and component-level design

This section will provide an outline of the various components and subsystems of the Online IQ Management System.

3.1 System Structure

This system will entail several modules, namely; Admin, Company, Test, Database and Applicant.

### 3.1.1 Architecture diagram

**Test**

- Questions.

- Guideline.

- Help manual.

**Company**

Creates an account and waits the account to be approved by Admin. But can approve applicant’s account.

- Set quiz.

**Database**

- Applicant’s Scores.

- Company’s Details.

- Applicant’s Details.

More Features

**Applicant**

Creates an account. Waits for the approval from either Company or admin. Takes the quiz

**Admin**

Approve Company’s account and also can approve applicant’s account if asked to do so by the company. Adds features to the system

## 3.2 Description of components

### 3.2.1 Description of login screen

|  |  |
| --- | --- |
| Identification | LoginScreen |
| Type | Form |
| Purpose | The login screen assures that only admin, company and applicants can access the system. |
| Subordinates | This screen contains links to the following screens:   1. Main Menu Screen 2. New User Account Screen |
| Dependencies | The following screen links to this screen:   1. Main Menu Screen |
| Interfaces | The links are contained in the bottom half of the screen. |
| Resources | Database Access Requirements: access to the applicant’scores found in the appropriate database tables. |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects. |
| Data | The data for the Login Screen is the username and password entered by the user. It is validated with a query against the database. |

### 3.2.2 Description of New User Account Screen

|  |  |
| --- | --- |
| Identification | NewUserAccountScreen |
| Type | Class/Form |
| Purpose | The new user account screen allows new companies or applicant’s the ability to create a unique user name for himself, which can be used to log into the system. |
| Subordinates | This screen contains links to the following screen:   1. Login Screen |
| Dependencies | The following screen links to this screen:   1. Login Screen |
| Interfaces | The links are contained in the bottom half of the screen. |
| Resources | Database Access Requirements: access to the Applicant/Company registration information found in the appropriate database tables. This access is used to create a new user and check to make sure that the provided information is legit. |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects. |
| Data | The data supplied by the system are fields the new user must enter. The data given by the user is the appropriate information needed to fill in the given fields. This data once determined valid, by checking to make sure the user does not already exist, is saved in the database. |

### 3.2.1 Description of main menu

|  |  |
| --- | --- |
| Identification | MainMenuScreen |
| Type | Class/Form |
| Purpose | The main menu screen assists the user by presenting them with tasks the program performs. |
| Subordinates | This screen contains links to the following screens:   1. Quiz Screen 2. Applicants score Screen |
| Dependencies | The following screens link to this screen:   1. Login Screen 2. Test Screen 3. Company Screen 4. Admin screen |
| Interfaces | The links are contained in the bottom half of the screen. The screen is designed to be easy to view using the resolution standard on the PDA. |
| Resources | None |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects. |
| Data | There is no data entered for this screen. |

## 3.3 Database Modelling

*Data Normalization*

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

*Entity Relationship Model*

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

Applicant 2

System

Aplicant 1

Company

Admin

IQ Test Sys

Systemm

Test

# 4.0 User interface design

The user interfaces via a text input screen. The user is prompted for several values in order to perform the calculations.

4.0.1 Expected Input:

The user is prompted to enter values for the following variables:

* Applicant’s/company’s Name
* Company’s details.
* Company set quiz.
* Answers to the quiz.

### 4.0.2 Output

The applicant will receive a report after taking the quiz indicating the score they have managed to score in the test and also tell the applicant if they meet the expected scores. If so, the system will tell the applicant that they qualify to the next step of the interview.

## 4.1 Objects and actions

### 4.1.1 Applicants use Case

The applicant has the following use case:

### 4.1.2 Company Use Case

Applicant

Take the IQ test

The company has the following sets of use cases

Register account

Company

Approve Applicant’s acc.

Set questions

Set pass scores

Set deadline

### 4.1.3 Admin use case

Approve Co. acc.

Admin

Add Features

Monitor DB

Approve Applicant’s acc.

Provide security

# 5.0 Restrictions, limitations, and constraints

The main identified limitation is that it will be impossible several companies to share website home page thus it will the company to host a whole module to online services. Restrictions in this system is that the applicant will not be able to access the services or the test if the their account is not approved by the named company. This program takes up about 7 kb of memory. The output reports are modest in size and take up

about 7 kb.

# 6.0 Testing Issues

No identified testing issues.

## 6.1 Classes of tests

Register

Logout

Quiz

login

Scores

Applicant

## 6.2 Expected software response

Expected response from the companies is the security restrictions in this system.

## 6.3 Performance bounds

This software is intended to be hosted on any UNIX webservers with no modifications needed to support different sites.

## 6.4 Identification of critical components

The database is a critical component since in the development the system will use SQL lite to store data. If there is too much data that will need to be saved at a go then this might result to full storage.

# References

Booch, Grady, Ivar Jacobsen, and James Rumbaugh. The Unified Software Development Process

(The Addison-Wesley Object Technology Series). 1st. Ed. New York: Addison Wesley, 1999..