

**Software Requirements Specification**

**For**

**Checklist Application**

**Version 2.0 approved**

**Prepared by**

**Midbrains Technologies**

**31 – 10 - 2018**

***Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

***Software Requirements Specification for <Project>*** ***Page ii***

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Table of Contents .....................................................................................................................** | | **ii** |
| **Revision History .......................................................................................................................** | | **ii** |
| **1. Introduction .........................................................................................................................** | | **1** |
| 1.1 | Purpose ....................................................................................................................................... | 1 |
| 1.2 | Scope of Project .......................................................................................................................... | 1 |
| 1.3 | Document Conventions .................................................................. | **Error! Bookmark not defined.** |
| 1.4 | Intended Audience and Reading Suggestions .................................. | **Error! Bookmark not defined.** |

|  |  |  |
| --- | --- | --- |
| **2. Overall Description ..............................................................................................................** | | **2** |
| 2.1 | Product Perspective ..................................................................................................................... | 2 |
| 2.2 | User Classes and Characteristics ................................................................................................. | 1 |
| 2.3 | Operating Environment ................................................................................................................ | 1 |
| 2.4 | Design and Implementation Constraints ......................................... | **Error! Bookmark not defined.** |
| 2.5 | Assumptions and Dependencies ................................................................................................... | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **3.** | **System Features ...................................................................................................................** | | **4** |
|  | 3.1 | System Feature 1 ......................................................................................................................... | 4 |
|  | 3.2 | System Feature 2 (and so on) ....................................................................................................... | 4 |
| **4.** | **External Interface Requirements ........................................................................................** | | **5** |
|  | 4.1 | User Interfaces ............................................................................................................................ | 5 |
|  | 4.3 | Software Interfaces...................................................................................................................... | 5 |
|  | 4.4 | Communications Interfaces .......................................................................................................... | 5 |
| **5.** | **Functional Requirements ....................................................................................................** | | **6** |
|  | 5.1 | Client/Server System ................................................................................................................... | 6 |
| **6.** | **Other Nonfunctional Requirements....................................................................................** | | **6** |
|  | 6.1 | Performance Requirements .......................................................................................................... | 6 |
|  | 6.2 | Security Requirements ................................................................................................................. | 6 |
|  | 6.3 | Software Quality Attributes ......................................................................................................... | 6 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

***Software Requirements Specification for <Project>*** ***Page 1***

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to present a detailed description of the Checklist Application. It will explain the purpose and features of the application, the interfaces of the system, what the system will do, the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

**1.2 Scope of Project**

This Software Application will be a Checklist Application for Admin. This application will be designed to maximize the User’s productivity by providing tools in the Checklist process. By maximizing the User’s work efficiency and production the system will meet the User’s needs while remaining easy to understand and use.

More specifically, this application software is designed to allow an Admin to manage a group of Users and to update user data. The software will facilitate Checklist Data between User and Admin. The system also contains a relational database containing a list of Users and Admin.

**1.3 DOCUMENT CONVENTIONS**

This document uses the following conventions. <<*Include the conventions as per your application* >>

|  |  |
| --- | --- |
| DB | Database |
|  |  |
| DDB | Distributed Database |
|  |  |
| ER | Entity Relationship |
|  |  |

***Software Requirements Specification for <Project>*** ***Page 2***

**1.4 INTENDED AUDIENCE AND READING SUGGESTIONS**

This project is a prototype for the Checklist Application and it is restricted within the client’s company premises. This has been implemented by developers of Midbrains Technologies. This project is useful for Admin and as well as to the User.

**2. Overall Description**

**2.1 Product Perspective**

A Checklist Application stores the following information.

* Checklist details:
  + It includes following features in User Panel –

-The Ability to start the checklist task by clicking start/run button

-The checklist page will have check list task set

-Ability for the user to start, pause and close checklist

* + - * Pop-up notification sent to user when checklist task not approved
      * Ability for user to enter data in checklist
      * Ability to validate data
* User Management:
  + It includes following features in Admin Panel –
    - Add/Edit/Update User
    - Activate/Deactivate User

***Software Requirements Specification for <Project>*** ***Page 1***

* Search Management:
  + It includes following features in Admin Panel –
    - Ability to search users based on Keywords
    - Ability to list searched user
    - Ability to export data in CSV format
* Role Management:
  + It includes following features in Admin Panel –
    - * Ability to assign roles to users
      * Ability to grant/deny roles to users
      * Ability to assign checklist task to user
* Category Management:
  + It includes following features in Admin Panel –
    - Add/Edit/Update Category
    - Ability to assign checklist to category
    - Activate/Deactivate Categories

***Software Requirements Specification for <Project>*** ***Page 1***

* Checklist Management:
  + - It includes following features in Admin Panel –
      * + Ability to create checklist items
        + Add/Edit/Update/Delete checklist items
        + Ability to clone and rename checklist items
        + Ability to manage instruction for items
        + Ability to manage response type in Checklist items
* Notification Management:
  + It includes following features in Admin Panel –
    - * Ability to send notification to user for upcoming and overdue tasks
      * Ability to send notification to admin after checklist submission
      * Ability to send notification to user for task overdue
      * Ability to send checklist status notification in real time

**2.2 User Classes and Characteristics**

Users of the system should be able to feed information in the Checklist Application and Admin can retrieve information through Admin Panel. Users can go through Checklist Section after login. And Admin can do User Management, Search Management, Role Management, Category Management, Checklist Management and Notification Management after login. The User/Admin should be able to do the following functions:

* User:
* Registration Page
* Register with Email ID/Username
* Set Password
* Email Verification
* Login Page - User Name

***Software Requirements Specification for <Project>*** ***Page 2***

* Password
* Forgot Password
* Checklist Section
* Admin
* Login Page - Username - Password
* User Management
* Search Management
* Role Management
* Category Management
* Checklist Management
* Notification Management

**2.3 Operating Environment**

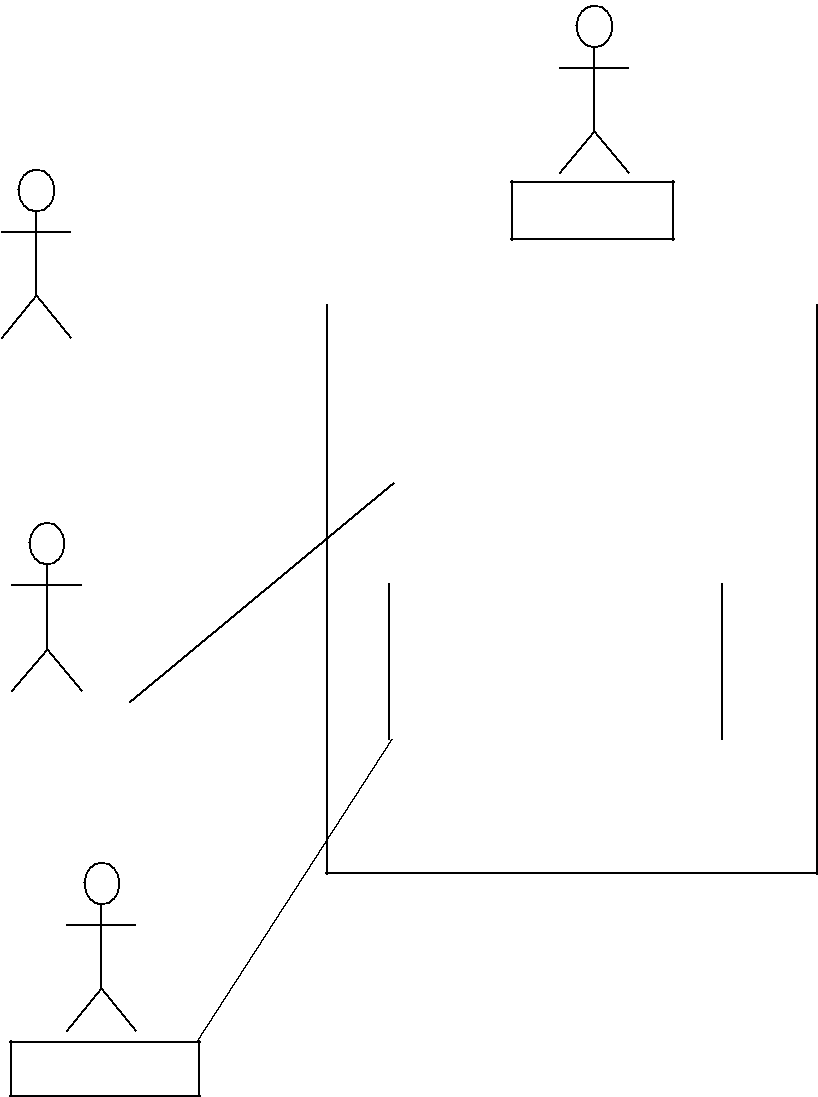
Operating environment for the Checklist Application is as listed below. <<*Include the details as per* *your application* >>

* C#
* ASP.NET
* MVC
* Database: SQL
* Web Services
* WCF REST AP

**2.4 User Design and Implementation Constraints**

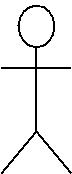
1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries.
4. Explain how various fragments will be combined to do so.
5. Implement the database at least using a centralized database management system.

***Software Requirements Specification for <Project>*** ***Page 3***



Admin

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| User |  |  |  |  | Checklists | | |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |



User Management

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | HS DB |  |
| User |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |
|  |  |  |  |  |  |  |  |

Checklist Management

User

*Fig. Use Case Diagram for Checklist Application (User and Admin)*

**2.5 Assumptions and Dependencies**

Let us assume that this is a distributed Checklist Application and it is used in the following application:

* A User will create his profile on Checklist Application, and he can Update his data on Checklist Section.
* An Admin can manage User Data, and he will be able to create Change/Update data. Assuming both the User and Admin, we will design a database that is dispersed for both

Admin and User.

***Software Requirements Specification for <Project>*** ***Page 4***

**3. System Features**

**3.1 System Feature 1**

The major features of e-administration of computer lab system are as listed below.

* Unauthorized access is prevented. Because only authorized user can address the complaints and access the resources.
* It helps the administrator to keep track of the detailed information of the labs such as maintaining each user’s profile, the number of available resources etc.
* It helps instructors to dump the software into the server to facilitate students to take the required software.
* It increases the availability of lab resources by their proper maintenance.
* A smooth communication link establishes between different users and user admin.

**3.2 System Feature 2 (and so on)**

The Checklist Application maintains information of Users, and Admin can see all the updated data of User. Of course, this project has a high priority.

By achieving above features, Responsiveness and hence outcomes of each lab increases.

**Modules**

1. **Role of admin**

Manages a user, assigns access rights, authenticates users, checks report, and assigns the task.

These functions are used to keep track of various activities in labs.

1. **Service complaint**

It includes complaint registration, assigning a task to the instructor, addressing complain. These functions may be used to register a complaint about improper conditions of devices (in the lab) to admin in order to get response/service.

1. **Software maintenance**

This module checks warranty of particular software generates alert messages and helps in proper documentation of available software in labs.

***Software Requirements Specification for <Project>*** ***Page 5***

**4. External Interface Requirements**

**4.1 User Interfaces**

* Front-end software: ASP.NET (4.0/4.5)
* Back-end software: SQL

**4.2 Software Interfaces**

Following are the software used for the Checklist application –

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | We have chosen Windows operating system for |
|  | its best support and user-friendliness. |
| Database | To save the User’s records, we have chosen |
|  | SQL+ database. |
| ASP.NET MVC | To implement the project we have chosen |
|  | ASP.Net language for its more interactive |
|  | support. |
| Web Services | To provide web services in the Project we have |
|  | chosen |

**4.3 Communications Interfaces**

In Checklist Application, User will send information with the help of Checklist Application and Admin will get User’s information with the help of Web Services to fetch information of User.

**5. Functional Requirements**

**5.1 CLIENT/SERVER SYSTEM**

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

* Some sites are client sites and others are server sites.
* All the data resides at the server sites.
* All applications execute at the client sites.

***Software Requirements Specification for <Project>*** ***Page 6***

**6. Other Nonfunctional Requirements**

**6.1 Performance Requirements**

The logical structure of the data to be stored in the internal Admin Manager database is

given below.

**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.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

**6.2 Security Requirements**

The server on which the Checklist Application resides will have its own security to prevent

unauthorized *write*/*delete* access. There is no restriction on *read* access. The use of email by an

Admin or User is on the client systems and thus is external to the system.

The PC on which the User Manager resides will have its own security. Only the Admin will

have physical access to the machine and the program on it. There is no special protection built into

this system other than to provide the Admin with *write* access to the Checklist Application to

manage User activities.

**6.3 Software Quality Attributes**

* **AVAILABILITY:** The data should be available for the specified user and specified time asmany users will create their accounts.

***Software Requirements Specification for <Project>*** ***Page 7***

* **CORRECTNESS:** The data should reach from correct User and should reach the Admin.
* **MAINTAINABILITY:** The administrators can easily maintain theuser’sdata.
* **USABILITY:** The Admin should satisfy a maximum number ofuser’sneeds.