**Software Requirement Specification (SRS) for**

**Highly Confidential Security System (HCSS)**

**1.Introduction**

In day-to-day life it is quite hard to remember all confidential data like Mail Id, Password, bank account number, Insurance policy number, PAN number, Driving License number, education certificate numbers, some highly value scan copy, some confidential photo, music and videos. HCSS is a highly secure web application to store all confidential data in single credential.

**1.1 Purpose**

HCSS is a web application developed for secure and easy access of data. HCSS is useful and convenient application that spares you the trouble of remembering the passwords and securing the files. This application helps people to store their passwords and various types of files like photos, music and videos in a secure and efficient manner. The application which we develop here uses state of the art encryption technology to secure files and access over anywhere in the world using the internet.

**1.2 Scope**

* Users are admin and public.
* All users have their own login page.
* Administrator has an ability to manage the registered users and organise the files stored in server.
* Administrator has an ability to provide the notification to the logged users.
* Public are the users using this web application to protect their important details in single credentials.
* Users can access their files through internet from anywhere in the world.

**1.3 Definitions, Acronyms, and Abbreviations**

**AJAX-** Asynchronous JavaScript and XML is about updating parts of a web page.

**UML-** Unified Modeling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document.

**XML-** eXtensible Markup Language is a text based format that let developers describe, deliver and exchange structured data between a range of applications to client for display and manipulation.

**JSP-** Java Server Pages is used to create dynamic web content.

**J2E-** Java Enterprise Edition is a programming platform which is a part of java platform for developing and running distributed java.

**HTTP-** Hyper Text Transfer Protocol is a transaction oriented client/server protocol between web browser and a web server.

**HTTPS-** Secure Hyper Text Transfer Protocol is a HTTP over SSL (Secure Socket Layer).

**1.4 References**

* Software Engineering Concepts and Practices-Ugrasen Suman.
* Database Management Systems-Navathe.
* Object Oriented Modelling and Design with UML- Michael Blaha
* Java Complete Reference- Herbert Schildt.
* Cryptography and Network Security- William Stallings.

**1.5 Overview**

The SRS will include two sections, namely:

Overall Description: This section will describe major components of the system, interconnections and external interfaces.

Specific Requirements: This section will describe the functions if actors, their roles in the system and the constraints faced by the system.

**2. Overall Descriptions**

**2.1 Product Perspective**

The HCSS is available for use by the admin and the user. The admin and users will use the webpage as a front end. The browser goes through an http server. Application server manages the connection between the front end and back end. All types of information and data that are necessary for the users are stored in database.

**2.2 Product Functions**

HCSS should support the following use cases:

|  |  |  |
| --- | --- | --- |
| **Class of Use Cases** | **Use Cases** | **Description of Use Cases** |
| Use case related to  installation | Installation | HCSS Installation |
| Use case related to  Sign in | Login | 1.User login to the account  2.Admin login to the account |
| Use case related to  Registration | Registration | Registration before going to login |
| Use case related to  Authentication of  Registration | Send mail | Mail verification  System verifying registration  System accepting genuine user |
| Check registration |
| Check name |
| Use case related to  Authentication of  password | Password token | System sending password token  User and admin are provided with validate password  Change of password |
| Check password |
| Change password |
| Use case related to  profiles | Profile management | Admin managing profiles  Updating profiles  Viewing data to users |
| Update profile |
| View profile |
| Use case related to  Notifications | View notification | Viewing notifications to users  Admin pushing notifications |
| Push notification |
| Use case related to  Reports | Generate reports | Admin generating reports |
| Use case related to  Lockers | Create locker | Creating locker  Renaming locker  Deleting locker |
| Rename locker |
| Delete locker |
| Use case related to  Blocking | Blocking abuse user | System has to block abuse users |
| Use case related to  Account | Renaming account | Renaming the existing account  Removing if not needed |
| Removing account |
| Use case related to  Coding and decoding | Encrypt/ Decrypt | Encryption and decryption |

* The HCSS service provides the user interface to user and admin for creating their profiles.
* This product has the feature of sending the "Account verification link" to the user's mail, while creating their account.
* This product has the ability to send "Password Reset link" to the user's mail, if the user forgets the password for accessing the account.
* This product provides the users to create the locker facilities for Bank account information, Music, Videos and Image files.
* This product allows the registered users to store their important details on a server in an encrypted format and while downloading, it is automatically decrypted.
* This product provides the facility for admin to generate the reports in user's logs in PDF and Microsoft Excel formats.
* It ask the account's current password from the user when they want their personal credentials from our server as a report.

**2.3 User Characteristics**

* Both users and admin on internet will be using HTTP protocol.
* Users use the FTP protocol for uploading and downloading the files from or to the server.
* Admin configures the SMTP Server for sending the mail to the user.

**2.4 General Constraints**

* GUI is only in English.
* Login and password is used for the identification of the user and admin.
* Limited to HTTP, SMTP and FTP protocols.
* Centralized server is used.

**2.5 Assumptions and Dependencies**

HCSS is a platform independent web application. It is assumed that the client or user computer has latest browser with JavaScript enabled. In addition, Firefox and Internet Explorer need the flash plug-ins to play the media files. For configuring the SMTP Server it needs the Linux operating system with at least kernel version 2.6.30, if server is in windows then it may need Windows Server. Sending the message to the users mainly depends on the network operator.

**3. Specific Requirements**

**3.1 Functional Requirements**

We describe the functional requirements by giving various use cases.

**Use case related to registration:**

**Use case 1:** Registration

Primary actor: Administrator

Pre-condition: Internet connection available.

Main Scenario:

1. The user accesses the registration page for new ID.

2. He/ She fills up the personal details like native place, age, gender etc. and submits.

3. The completeness of data is checked on user side.

4. The Database is updated.

Alternate Scenario:

1. The data completeness check fails and the user is prompted to provide all

details.

2. The database update fails.

**Use case related to system authentication:**

**Use case 2: User Verification**

Primary Actor: Administrator

Pre-Condition: Internet Connection availability

Main Scenario:

1. Starts the application and checks the provided name with already existing

name during registration.

2. User gives the login password.

3. System does authentication.

4. Main screen is displayed.

Alternate Scenario:

1. Authentication fails.

2.Promt the user that he typed the wrong password.

3. Allow him to re-enter the password. Give him 3 chances.

**Use case related to signing in:**

**Use case 3:** Login

Primary Actor: user

Pre-Condition: Internet connection availability

Main Scenario:

1. User starts the application and enters username and password.

2. System verifies the input.

3. User's respective home page is displayed.

Alternate Scenario:

1. User may enter wrong username or password.

2. Allows him to re-enter data.

3. Providing 3 chances to enter login data.

**Use case related to security:**

**Use case 4:** Creating Security

Primary Actor: Administrator

Pre-Condition: User logged in

Main Scenario:

1. User selects the data in which the security is to be created.

2. User initiates the "create security" functionally.

3. System asks the user to enter the attributes of the security.

4. User specifies the following fields:

a. Name of the locker

b. Type of the location file/ folder/ zip

5. An empty security of specified attributes is created.

Alternate Scenario:

1. A security with the given name have already exists>

2. Security creation fails, error message is displayed.

**Use case 5:** Rename security.

Primary actor: user

Pre-condition : user logged in

Main scenario:

1. User selects the data in which the security is to be created.

2. System asks for the password to be renamed and the new name.

3. User enters the new name.

Alternate scenario:

1. The user whose name is supposed to change does not exist.

2. Renaming fails, the error message is displayed.

3. security with the same new name exists.

Use case 6: Delete security

Primary actor: user

Pre-condition: user logged in

Main scenario:

1. User initiates the delete user functionality.

2. System asks for the name of the user.

3. The password is deleted.

Alternate scenario:

1. security does not exist.

2. Deletion fails, error message is displayed.

User case 7: Display security

Primary actor: user

Pre-condition: user logged in

Main scenario:

1. User selects the data in which the security is to be created.

2. User initiates the create security functionality.

3. System asks the user to enter the attributes of the security.

Use Case 8: Setting alerts

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario :

1. User initiates the set alert functionality.

2. The system asks the user for the date and details of the alert.

3. The alert is set.

Use Case 9: Change Password

Primary Actor: User

Pre Condition: User logged in

Main Scenario:

1. User initiates the password change command.

2. User is prompted for old password, new password and confirm password

3. User gives the old password, new password and confirm new password

4. System does authentication.

5. New password is registered with the system.

Alternate Scenario :

1. Authorization fails

2. Prompt the user that he typed the wrong password

3. Allow him to re-enter the password. Give him 3 chances.

4. New password and confirm new password do not match.

5. Allow him to re-enter the attributes. Give 3 chances.

Use Case 10: Delete username.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario :

1. User initiates the user name functionality.

2. User is asked for the name to be deleted.

3. The user name is deleted.

Use Case 11: Delete password

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario :

1. User initiates the user name functionality.

2. User is asked for the password to be deleted.

3. The password is deleted.

**3.2 External Interfaces Requirements**

**3.2.1 Administrators**

Database Management: Control the database and keep track of all the details of the users who were registered in the HCSS system.

Preconditions: Administrator is already logged in.

Main Scenario:

* Normal check of the database by the Administrator.
* Updating the database if required.
* The updating will be done only by the user with the permission of the admin.

Post Condition: Always update database

**3.2.2 Giving Permission to users**

Allow with the users and give permission to access their data in the site after checking whether the user is valid user or not.

Preconditions:

* Administrator is already logged in.
* User access the data from admin.

Post Conditions: Possibilities of updating the user details by the user.

**3.3 Performance Requirements**

* 24/7 availability
* High speed
* Compatibility for all the Web Browsers
* Match able to most of the file types
* Wide listening platforms and compatible with players

**3.4 Design Constraints**

Database:

The system shall use the SQL Database.

Operating System:

The Development environment shall be Windows versions, vista.

Web-Based:

The system shall be a web-based application.

**3.5 Security Requirements**

Passwords will be saved encrypted in the database in order to ensure the user's privacy.

The user's IP will be logged.

The details of the users will be stored in the server database maintained by the admin can be viewed and modified by the user.

The user will grant to access the data by the admin. He can modify the details.

In some cases the user will be forget his details about his particulars he can retrieve it from the admin.

Data integrity will be checked for critical variables.

**3.6 Maintainability Requirements**

A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization to the program will be done. Also the software design is being with modularity in mind so that maintainability can be done efficiently.

**3.7 Reliability Requirements**

The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Also the system will be functioning inside a container. Thus the overall stability of the system depends on the stability of container and its underlying operating system.

**3.8 Availability Requirements**

Secure access to all the confidential data.

24/7 availability.

Network connection is required for all the users.

**3.9 Data base Requirements**

Data base should only accept consistent data.

Updating must be done properly.

Crashing of data must be avoided.

Use of graphical user interface must be done well to show strategic data to admin.

**3.10 Documentation Requirements**

Simplicity of Interface.

Providing printing facility for users from history.

**3.11 Safety Requirements**

System use shall not cause any harm to human users.

The database may get crashed at any certain time due to virus or opening system failure. Therefore it is required to take the database backup.

**3.12 Operational Requirements**

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. Users have to be made aware that the general idea of operating the application must be known before he uses the system.