**COMSATS University Islamabad,**

**Abbottabad Campus**

**SOFTWARE REQUIREMENTS SPECIFICATION   
(SRS DOCUMENT)**

**for**

**PAPERLESS RECORDS SHARING VIA FINGERPRINTS**  
Version 1.0

***By***

**Mohsin Munir CIIT/SP19-BSE-021/ATD**

**Bushra Mukhtar CIIT/SP19-BSE-023/ATD**

**Umar Farooq Hassan CIIT/SP19-BSE-027/ATD**

***Supervisor*MUHAMMAD TARIQ BALOCH**

***Bachelor of Science in Computer Science (2019-2023)***

**Table of Contents**

**Revision History 3**

**1.** **Introduction 5**

1.1 Purpose 5

1.2 Scope 5

**2.** **Overall description 5**

2.1 Product perspective 5

2.2 Operating environment 5

2.3 Design and implementation constraints 5

**3.** **Requirement identifying technique 6**

3.1 Use Case Diagram 6

3.2 Use Case Description 6

**4.** **Specific Requirements 8**

4.1 System feature X 8

**5.** **Quality attributes** Error! Bookmark not defined.

5.1 Usability 9

5.2 Performance 9

**6.** **External interface requirements 35**

6.1 User interfaces 35

6.2 Software interfaces 35

6.3 Hardware interfaces 35

6.4 Communications interfaces 35

**7.** **Project Gantt chart 35**

**8.** **References 10**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Supervisor’s Name>**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

Paperless record saving is server-based system. In which a user can easily save their records. This System can also be referred as keyless authentication unlike traditional way where it needed a password to enter. With the help of this application, User needs to register first with basic registration details and needs to create valid login credentials in order to access the application. After successful creation of account, users can login using valid credentials and access the application. There will be list of documents where user can Add, Share, Update and Delete Documents. While Adding Documents, the Documents will be verified by the Certificate Parameters from the Database. Documents are captured and can be resized or cropped to get the correct edges.

**Abstract:**

Paperless record saving is an Android based system it means one can store any type of file/notes on server through application. This is an advance system which provides the facility to store and retrieve the files without any duplication or if there is it alerts the user for further action. By using this application, we can reduce the use of paper. Furthermore, in these days Device storage is a major issue for Smart Phone User because we get limited storage which is either covered by multimedia or other apps which lacks space for important stuffs. This System can be used as private files/notes or personal diary or important notes; can be given multiple names but plays a similar role of recording notes and keeping it away from everyone then the phones owner. If there is no Biometric feature on the phone, this app can’t be used.

**Purpose**

* This project enhance the old traditional way of business and trade files/media sharing through technology.
* To Provide instant access to user.
* This application will be implemented with latest versions of libraries.
* No need of cash for exchange but each delivery will be charged according to distance.

**Scope**

* The project is divided into two parts
* Web site for administration purpose
* Mobile application for user.
* The user can access it from worldwide but we are limiting this to a limited to country.
* It give opportunity to those users who want to share their files and secure them on our servers.
* App usage will be free
* The user can also check media delivery report.

**Overall description**

**Product perspective**

Describe the product’s context and origin. Is it the next member of a growing product line, the next version of a mature system, a replacement for an existing application, or an entirely new product?

Our app is designed for users to save their personal/ business related files or media, and can download anytime they want.

**Operating environment**

### Web-server:

For management website for admin and franchise manager we will need to make it live on server, as for server we require Linux OS for example:

1. Debian-10 for Django and REST-framework.
2. Nginx server for Django-backend
3. Gunicorn for hosting front-end files for Django.

### Mobile-App:

To manage android app or iOS app we need 2 services where we can upload our mobile application.

1. Google play store for android app
2. Apple store for iOS app.
3. Minimum target android version should be 8 or greater.

### API:

For interaction between web-server and mobile application there must be an API to get and send response. We need same platform as web-server to upload APIs.

For performance measuring and testing we can use ‘Postman’ a tool to test APIs stand-alone or dependent APIs.

**Design and implementation constraints**

**Spiral Model:**

For our FYP we need to implement spiral model because it is iterative also, we need to plan for 30%, 60% and 100% and complete SDLC (software Development Life Cycle).

Planning is important for implementation and spiral model is useful because we need to delivery project in parts to supervisor and each iteration contains complete SDLC.

**Requirement identifying technique**

As our project’s major part is already implemented in real life, we need to study articles from multiple scholars available on google.

**Use case diagram:**

**Diagram

Description automatically generated**

**Use case description**

**Table 1 Show the detail use case template**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1 |
| **Use Case Name:** | Register |
| **Actors:** | Customer |
| **Description:** | Fill basic information required to use APP |
| **Trigger:** | Create profile button clicked |
| **Preconditions:** | PRE-1. Customer is new user into APP.  PRE-2. Customer should have verified email. |
| **Postconditions:** | [Describe the state of the system at the conclusion of the use case execution.  POST-1. Send signal to Database to update values.  POST-2. Enable customer to further verify the profile. |
| **Alternative Flows:** | * Go-to dashboard * Only View products |
| **Assumptions:** | * App is connected with server. * Register page should be opened. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC-2 |
| **Use Case Name:** | Add Profile |
| **Actors:** | Customer |
| **Description:** | Verified personal information for system to verify authenticity. |
| **Trigger:** | Create profile button clicked |
| **Preconditions:** | PRE-1. Customer is logged into APP.  PRE-2. Customer should have verified email. |
| **Postconditions:** | [Describe the state of the system at the conclusion of the use case execution.  POST-1. Send signal to Database to update values.  POST-2. Enable customer to upload a product. |
| **Alternative Flows:** | * Go-to dashboard * Only View products |
| **Assumptions:** | * Customer is logged in. * Profile settings page should be opened. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3 |
| **Use Case Name:** | Upload Media |
| **Actors:** | Customer |
| **Description:** | New Media that a customer wants to Share with someone. |
| **Trigger:** | Create profile button clicked |
| **Preconditions:** | PRE-1. User must be verified user.  PRE-2. A category must be selected. |
| **Postconditions:** | POST-1. Send signal to Database to update values.  POST-2. Enable customer to upload a product. |
| **Alternative Flows:** | * Go-to dashboard * Only View products |
| **Assumptions:** | * Customer is logged in. * Profile settings page should be opened. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4 |
| **UseCase Name:** | Exchange order |
| **Actors:** | AppUser |
| **Description:** | User can select an equal valued product only to exchange the item. |
| **Trigger:** | Create profile button clicked |
| **Preconditions:** | PRE-1. Product must be selected  PRE-2. Product must be of equal estimated price.  PRE-3. Product must be verified by manager. |
| **Postconditions:** | POST-1. Send signal to Database to update values.  POST-2. Enable customer to start chat |
| **Alternative Flows:** | 1. Go-to dashboard 2. Chat with customer |
| **Assumptions:** | 1. App is connected with server. 2. Register page should be opened. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7 |
| **Use Case Name:** | Track shared Files |
| **Actors:** | AppUser |
| **Description:** |  |
| **Trigger:** | View history button clicked |
| **Preconditions:** | PRE-1. Customer is new user into APP.  PRE-2. Customer should have verified email. |
| **Postconditions:** | [Describe the state of the system at the conclusion of the use case execution.  POST-1. Send signal to Database to update values.  POST-2. Enable customer to further verify the profile. |
| **Alternative Flows:** | 1. Go-to dashboard 2. Only View products |
| **Assumptions:** | 1. App is connected with server. 2. Register page should be opened. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC-9 |
| **Use Case Name:** | Report A User |
| **Actors:** | Customer |
| **Description:** |  |
| **Trigger:** | Create profile button clicked |
| **Preconditions:** | PRE-1. Customer is new user into APP.  PRE-2. Customer should have verified email. |
| **Postconditions:** | [Describe the state of the system at the conclusion of the use case execution.  POST-1. Send signal to Database to update values.  POST-2. Enable customer to further verify the profile. |
| **Alternative Flows:** | 1. Go-to dashboard 2. Only View products |
| **Assumptions:** | 1. App is connected with server. 2. Register page should be opened. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC-10 |
| **Use Case Name:** | Receive Uploaded Media |
| **Actors:** | Franchise Manager |
| **Description:** |  |
| **Trigger:** | Create profile button clicked |
| **Preconditions:** | PRE-1. Customer is new user into APP.  PRE-2. Customer should have verified email. |
| **Postconditions:** | [Describe the state of the system at the conclusion of the use case execution.  POST-1. Send signal to Database to update values.  POST-2. Enable customer to further verify the profile. |
| **Alternative Flows:** | 1. Go-to dashboard 2. Only View products |
| **Assumptions:** | 1. App is connected with server. 2. Register page should be opened. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC-11 |
| **Use Case Name:** | authenticate product |
| **Actors:** | Franchise Manager |
| **Description:** |  |
| **Trigger:** | Create profile button clicked |
| **Preconditions:** | PRE-1. Customer is new user into APP.  PRE-2. Customer should have verified email. |
| **Postconditions:** | [Describe the state of the system at the conclusion of the use case execution.  POST-1. Send signal to Database to update values.  POST-2. Enable customer to further verify the profile. |
| **Alternative Flows:** | 1. Go-to dashboard 2. Only View products |
| **Assumptions:** | 1. App is connected with server. 2. Register page should be opened. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC-16 |
| **Use Case Name:** | Manage Reported accounts |
| **Actors:** | Super Admin |
| **Description:** |  |
| **Trigger:** | Create profile button clicked |
| **Preconditions:** | PRE-1. Customer is new user into APP.  PRE-2. Customer should have verified email. |
| **Postconditions:** | [Describe the state of the system at the conclusion of the use case execution.  POST-1. Send signal to Database to update values.  POST-2. Enable customer to further verify the profile. |
| **Alternative Flows:** | 1. Go-to dashboard 2. Only View products |
| **Assumptions:** | 1. App is connected with server. 2. Register page should be opened. |

**Functional Requirements**

Assuming that our customer can be buyer and seller at the same time as we are considering it as exchange.

* System should include two roles to share the files.
* **AppUser** shall be able to upload the pictures, video, PDF, etc. any kind of document of their items from their own account.
* The entrusted users can contact and communicate through chat.
* AppUser should be able to remove his fingerprint.
* AppUser can add up to 3 fingerprints.
* Thumb Print should be the main fingerprints.
* AppUser can only share files with added users.
* AppUser will be able to lock files.
* Only locked files should require fingerprint while transferring data.
* AppUser shall be able to pay shipping charges through e-wallet.
* AppUser shall be able to cancel order with-in 6hrs.
* AppUser should be able to report fraud, wrong Files, report user.
* System must notify **receiver AppUser** about new File Shared.
* Product upload should be notified to the nearest franchise in the area.
* Franchise manager must view product through images, description and videos.
* System must notify franchise manager about upcoming products.
* Franchise manager will have to mark order as sent with **deliverer** name.
* Super Admin can view reports of whole system or specific AppUser.
* Super Admin must create franchise manager’s account.
* Super Admin should be able to receive reports from AppUsers.
* Super Admin must take action on reports and update their status.
* Super Admin can view ongoing exchange of files between users.

**Non Functional Requirements**

**Usability**

Usability requirements deal with ease of learning, ease of use, error avoidance and recovery, efficiency of interactions, and accessibility. The usability requirements specified here will help the user interface designer create the optimum user experience.

* **ease of use:**

1. Interface should be simple and use symbols and icons for buttons.
2. Provide help text in forms and different fields.

* **error avoidance:**

1. To avoid errors done by user, system should be strong enough to cover validation checks.

* **Recoverability:**

1. Server must save Database copy once in 24hrs.

* **efficiency:**

Each API call and its execution should be done in 1-2 seconds.

**Capacity**

Our system files will need at least 500mb space on host-machine, and for database we will need minimum 20GB/year.

**Availability**

Database and APIs should be uploaded on a dedicated server which will be running 24/7, for users to access application anytime.

**Performance**

As it is referred as processing speed, throughput, and utilization… System should be able to:

1. Handle 20,000 requests per second.
2. Video and images should be stored on cloud storage for high quality and solves storage issue for our server.

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

**Deployment Diagram**

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated