

**SOFTWARE REQUIREMENT SPECIFICATION  
CARDZO  
DOCUMENT SCANNER APP**

Submitted By

**Sidarth CS  
Praveen Kumar T  
Krishnendu Suresh  
Jyothis KS  
Jabirhusain KP**

On

**20 - SEPTEMBER – 2020**

# **TABLE OF CONTENTS**

## **1. Introduction**

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Scope
- 1.4 Need For The System
- 1.5 References

## **2. Overall Description**

- 2.1 Product Perspectives
- 2.2 Product Features
- 2.3 User Classes And Charecteristics
- 2.4 Design and Implementation Constraints
- 2.5 Dependancies

## **3. System Features**

- 3.1 Functional Requirements

## **4. External Interface Requirements**

- 4.1 User Interfaces
- 4.2 Hardware Interfaces
- 4.3 Software Interfaces
- 4.4 Communications Interfaces

## **5. Non Functional Requirements**

- 5.1 Perfomance Requirements
- 5.2 Security and Privacy Requirements
- 5.3 Software Quality Attributes

# **1. Introduction**

## **1.1 Purpose**

The purpose of this Systems Requirements Specifications document (henceforth referred to as SRS) is to provide the guidelines necessary to design and implement software that fulfills all the requirements given.

In the case of our document scanner application (simply referred to as CARDZO ), this document will contain enough information so that in the event that the current team is not able to complete the software, a future team can use this document and only this document to create another version of the software that still fulfills all the requirements.

## **1.2 Document Conventions**

This Document was created based on the IEEE template for System Requirement Specification Documents.

- Glossory
  - RAM – Random Access Memory
  - UI – User Interface
  - SDK – Software Development Kit
  - App – Application
  - User – Person Who Uses The Application

## **1.3 Scope**

The app is mainly for those who intend to store their valuable documents in digital format and require a stable medium for that. Cardzo app will help them to scan the digital copies of every document, and store it on their android smartphone.

## **1.4 Need For The System**

- There are few apps which helps to scan the documents
- However, users need a simple application to scan and store their valuable documents in a secure way.
- Simple UI and avoiding non-essential features makes the usage of application more convenient.
- Also, Cardzo app stores the documents in local storage in a secured format.

## 1.5 References

1. <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>
2. <https://apro-software.com/writing-specifications-for-a-mobile-app-development-project/>
3. <https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/>

## **2. Overall Description**

### **2.1 Product Perspective**

The Cardzo app will ensure all the documents and user details are stored in the local database, and makes the retrieval of information or any documents possible. Even though there are some applications in the market with same functionalities, the complex user interface and non essential features makes the apps pretty useless for ordinary people. Here, through Cardzo app, the simplest user interface and essential features will make user's life easier.

### **2.2 Product Features**

- Set Up Profile – User need to set up a profile before start using the app.
- List Documents – There will be a list, in which user can view the scanned documents.
- Scan Document – User can a document when ever they want.
- Add Document – User can add documents from a photo, which is already in the gallery.
- Share Documents – User can share the documents to other app or save it to file.

### **2.3 User Characteristics**

- Types Of Users -
  - Android Smartphone Users
- Technical Expertise -
  - Users needs to have minimal knowladge of using smartphones and smartphone camera.

### **2.4 Design And Implementation Constraints**

- Platform :
  - Should work on Android Phones.
- Operating System :
  - Android – Version 6.0 above ( API 23 above )

- Hardware :
  - Smartphone should have camera
  - at least 1GB RAM

## **2.5 Dependancies**

- Our app makes use of a plug in ( Scanbot SDK flutter Plug-In)

## **3. System Features**

### **3.1 Functional Requirements**

- **Application will be initialized with user details.**
  - **Description** : The basic details of the user collected and stored in database.
  - **Implementation** : A UI with a simple form is launched on first launch. Then takes user inputs.
  - **Criticality** : This is crucial. Because user is the heart of our application.
  - **Risks** : The user may enter invalid data.
  - **Dependancy** : Relies on information from user.
- **Application will be able to retrieve previous documents scanned.**
  - **Description** : Shows a list of documents, which is already scanned and stored by the user.
  - **Implementation** : UI with a list of names and thumbnails, in which user can select a document.
  - **Criticality** : The main functionality of our app is storage and retrieval of documents.
  - **Risks** : The documents should be sorted using user's preference.
  - **Dependancy** : Depends on the local database by app.
- **Application will be able to scan documents**
  - **Description** : Using camera, the app will capture the photo of document and process the document to a much better digital form.

- **Implementation** : Using scanbot SDK, there will be a frame for showing the camera visuals and a button which have the functionality of scanning document.
  - **Criticality** : Crucial. The user must be able to add new documents to database.
  - **Risks** : Since, we rely upon an external plugin. There might be chances of unexpected behaviours.
  - **Dependancy** : Depends upon the Scanbot SDK flutter plug-in
- **Application will be able to display the documents**
    - **Description** : User can view the scanned document inside the application.
    - **Implementation** : A full screen UI with some buttons for navigation. The document will be displayed in a frame.
    - **Criticality** : Crucial. User should be able to view the documents.
    - **Risks** : There may be problem with data unavailability, due to the formatting or deleting of files.
    - **Dependancy** : Depends upon device local storage.
  - **Application will be able to add pre-captured photos from gallery app.**
    - **Description** : The application can scan and add documents to app from a photo available in the device.
    - **Implementation** : User should select the photo and the scanning and processing of document is done. Then it will be added to the app and stored in device.
    - **Criticality** : Required. Sometimes user will have old photos of documents. In that case, user can add the photos into app.
    - **Risks** : The photo format must be supported by app.

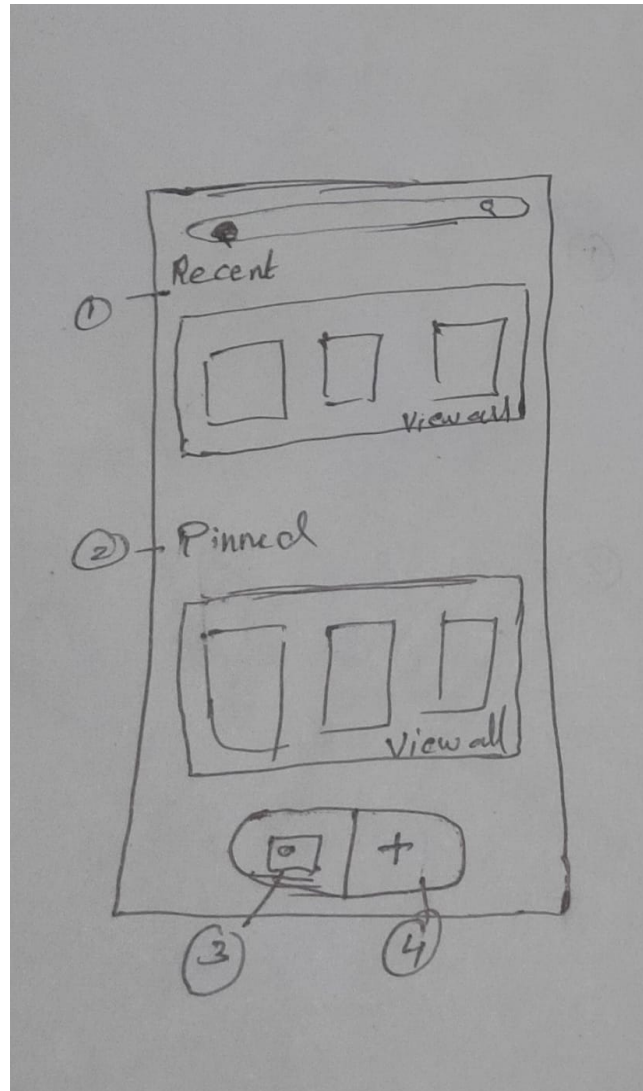


- **Dependancy** : Relies upon gallery app of android device.
- **Application will allow user to control the documents.**
  - **Description** : The user can rename and delete documents if they want. Also they can organize the documents.
  - **Implementation** : User can select a document and there will be a menu with the options for deletion and renaming.
  - **Criticality** : Crucial. Users needs to organize and delete or rename the documents as they want.
  - **Risks** : There may be an of accidental operation.
  - **Dependancy** : Relies upon user inputs.

## 4. External Interface Requirements

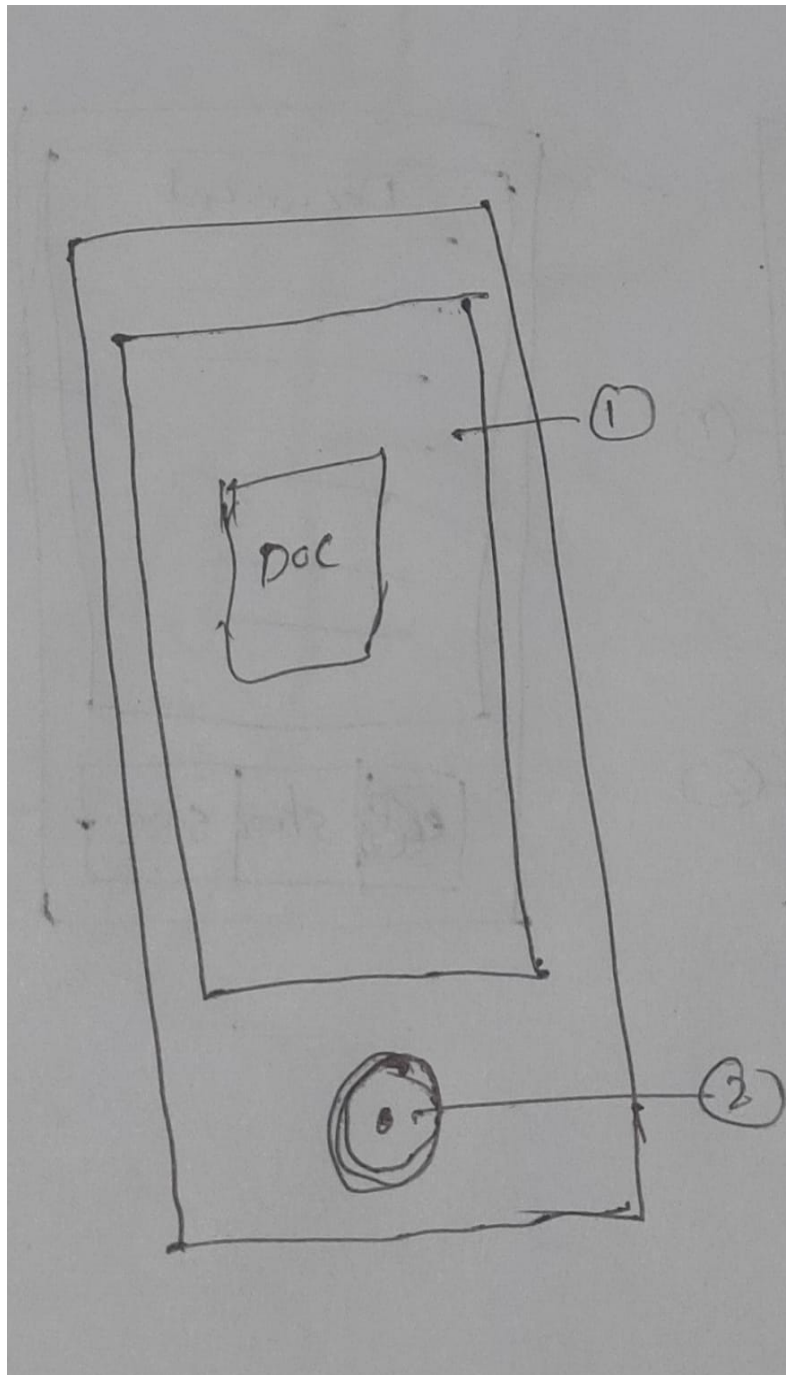
### 4.1 User Interfaces

- Main Activity



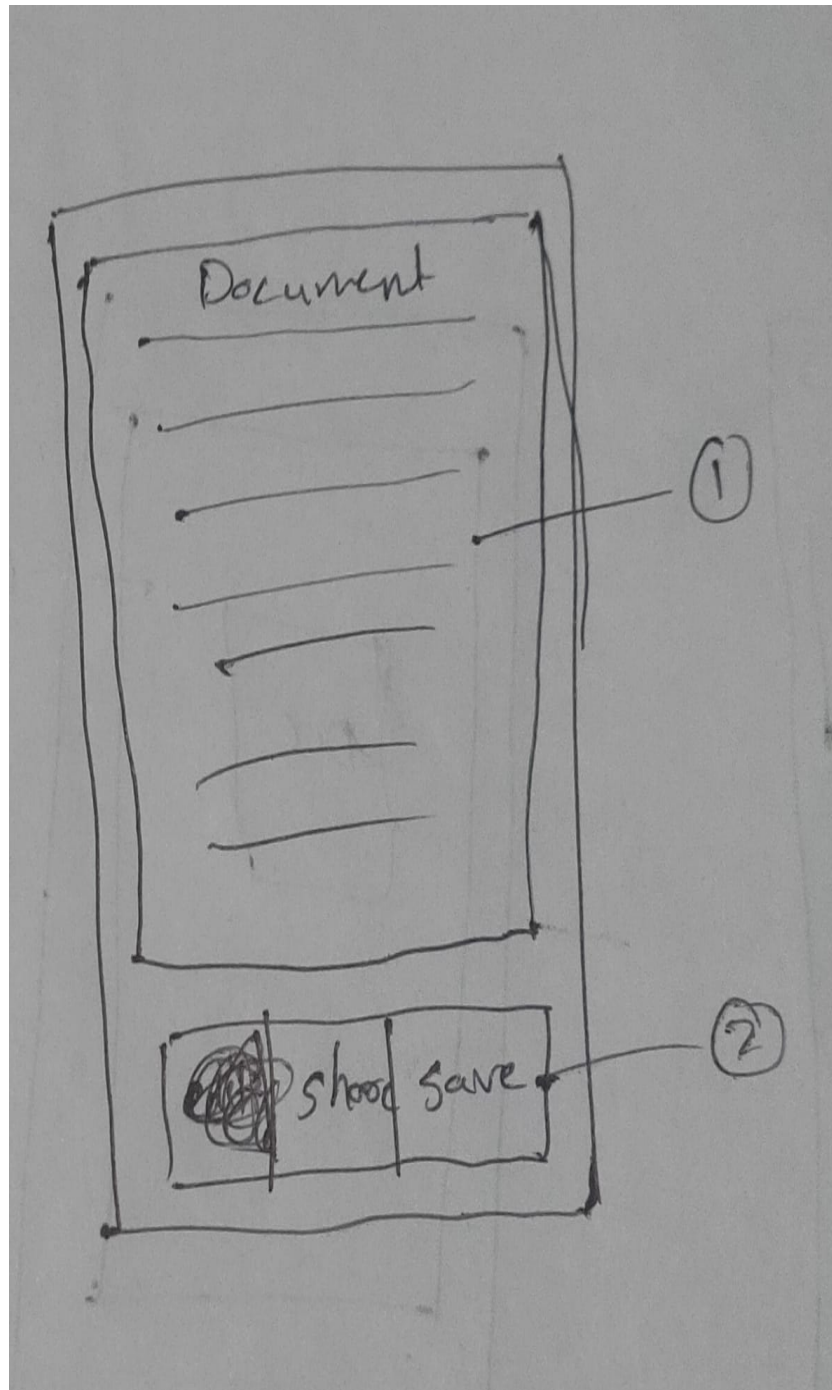
- |                 |  |
|-----------------|--|
| 1 . Recent      | : Shows the documents scanned recently |
| 2 . Pinned      | : Shows the user pinned documents      |
| 3 . Scan Button | : Goes to scanner activity             |
| 4 . Add Button  | : Goes to select activity              |

- **Scanner Activity**



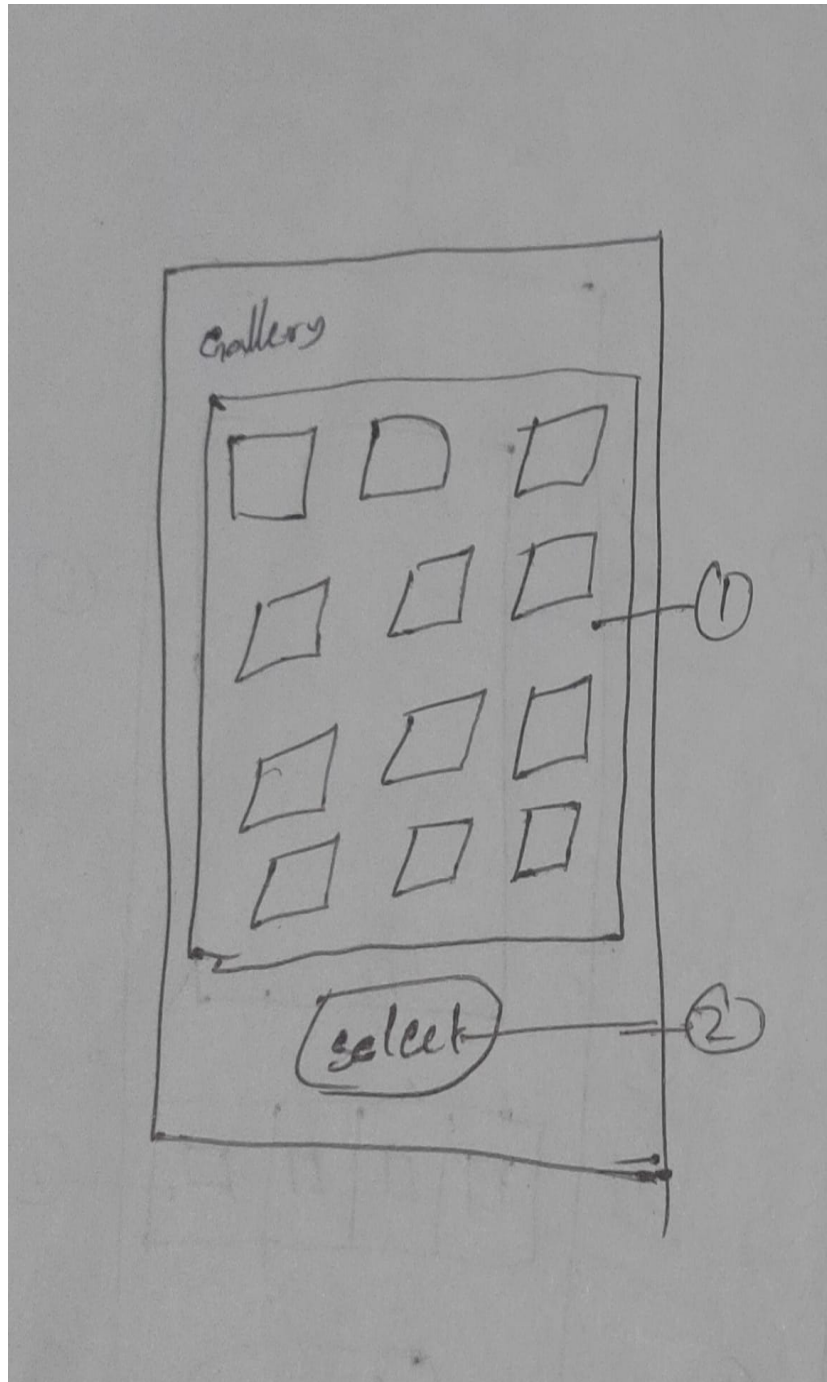
1. Scanning Frame : Shows the camera visuals
2. Scan Button : Starts the scanning process

- **Document Activity**



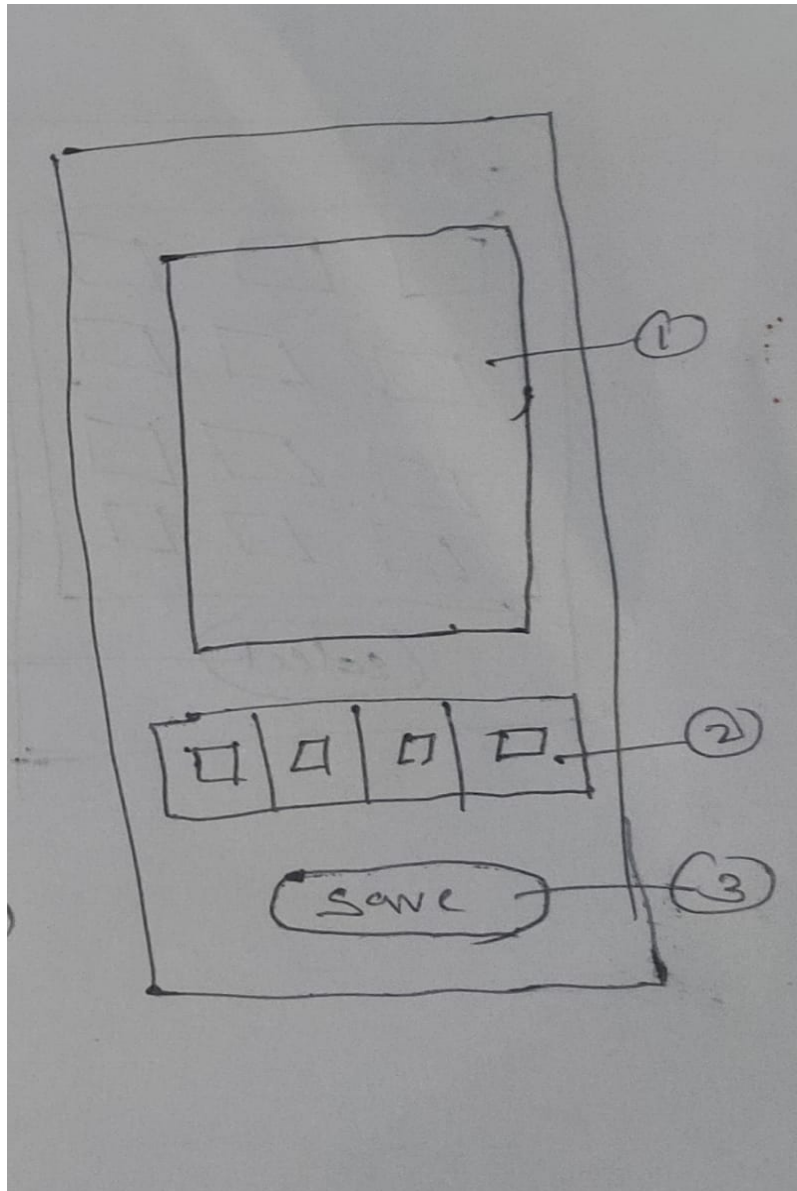
1. Document Viewer : Displays the document.
2. Toolbar editing. : Tools with different options like sharing and editing.

- **Selection Activity**



1. Gallery : Shows the images from gallery app.
2. Select Button : Selects the image for document scanning.

- **Process Activity**



1. Document Viewer : Shows the document in frame.
2. Toolbar : Tools for enhancement
3. Save Button : Saves the documents to local storage.

## **4.2 Hardware Interfaces**

- Smartphone with Android OS
- Camera for capturing the images

## **4.3 Software Interfaces**

- The scanner component will use the plug-in Scanbot SDK
- The gallery app is used for importing documents from existing images in gallery

## **4.4 Communication Interfaces**

- Internet – We need internet to connect with the scanning API.

## **5. Non Functional Requirements**

### **5.1 Performance Requirements**

This section describes the expected performance requirements. This is an estimation of the system, and all the numerical values may vary depends on how large the final application is.

- **Static Numerical Requirements**
  - The application shall support only one simultaneous user on each machine.
  - The application shall run on any android phone with android version 6 or above and with RAM more than 2 GB.
- **Dynamic Numerical Requirements**
  - The application shall be loaded and functioning within 10 seconds 95% of the time after starting the application.
  - Each Documents shall be processed within 10 seconds 90% of the time after scanning.

### **5.2 Security and Privacy Requirements**

Our application has a privacy and security sensitive nature, specifically with regards to the scanned documents of individuals.

In order to build trust with the users of our system, the system can make use of the following strategies:

- Encryption – Each and every documents are stored in encrypted format.
- Giving Control of Data to Users ( delete from the app whenever they want )

### **5.3 Software Quality Attributes**

- Reliability – The system will be able to handle the delays over internet and invalid inputs.
- Security – The system will use device's default security. Since, storage of user data is done, our application will implement its own security features like encryption.
- Maintainability – The application stores every data in local databases. Each components are mostly modular and independent of others.