# Artefact

## FDD

## SRS Legend

Sub Systems:

MCS : Mathematical Calculation System

CRS : Character Recognition System

HCS : Handwritten Character Recognition System

IPS : Image Processing System

MCS - F - 1.0

Types of Requirements:

F : Functional Requirements

NF : Non-Functional Requirements

UR : Usability Requirements

Numbering

**Legend**

## SRS Table

|  |  |
| --- | --- |
| **Requirement Code** | **Requirement Description** |
| MCSF **1.0** | The system should use the built in ALU to perform arithmetic calculations. |
| MCSNF 1.1 | The system should provide the result in less than 1.5 seconds. |
| MCSNF 1.2 | The system should follow BODMAS rule for calculations. |
| MCSUR 1.1 | The system should have “All Clear” button to reset the entry. |
| MCSUR 1.2 | The system should have “Clear Entry” button to erase the latest entry. |
| MCSUR 1.3 | The system should display result in bigger font than the input. |
|  |  |
| MCSF **2.0** | The system should be able to perform scientific calculations. |
| MCSNF 2.1 | The system should be able to point out the error if any occurs. |
| MCSNF 2.2 | The scientific notations should be placed separately. |
|  |  |
| MCSF **3.0** | The system should keep records of the calculations history. |
| MCSNF 3.1 | The system should display maximum five history records. |
| MCSNF 3.2 | The system should have clear history option. |
| MCSUR 3.1 | The system should have black background with white text color. |
|  |  |
| CRSF **1.0** | The system should recognize numbers as well as mathematical notations. |
| CRSUR 1.1 | The notations should be clear and familiar to the users. |
| CRSF **2.0** | The system should recognize characters with noises as well. |
| CRSNF 2.1 | The system should recognize the characters in less than 2 seconds. |
|  |  |
| HCSF **1.0** | The system should recognize hand written characters. |
| HCSNF 1.1 | The system must have handwritten accuracy over 80%. |
|  |  |
|  |  |
| IPSF **1.0** | The system should be able to capture image to perform calculations. |
| IPSNF 1.1 | The system should use mobile camera to capture the image. |
| IPSNF 1.2 | The system should capture the in 720\*720 resolution. |
| IPSNF 1.3 | After capturing the image there should be crop option. |
| IPSNF 1.4 | There should be save option for image. |
| IPSUR 1.1 | There should be camera icon to switch between standard and scanning calculator. |
|  |  |
| IPSF **2.0** | The system should be able to perform calculations on images from gallery. |
| IPSNF 2.1 | While importing, the images should be shown from latest to old. |
| IPSNF 2.2 | While importing only one image should be selectable. |
| IPSUR 2.1 | The selected image should be faded. |
|  |  |

## System Modeling

### Context Modeling

#### Use Case Diagram

* Use Case Diagram for Basic Calculation System

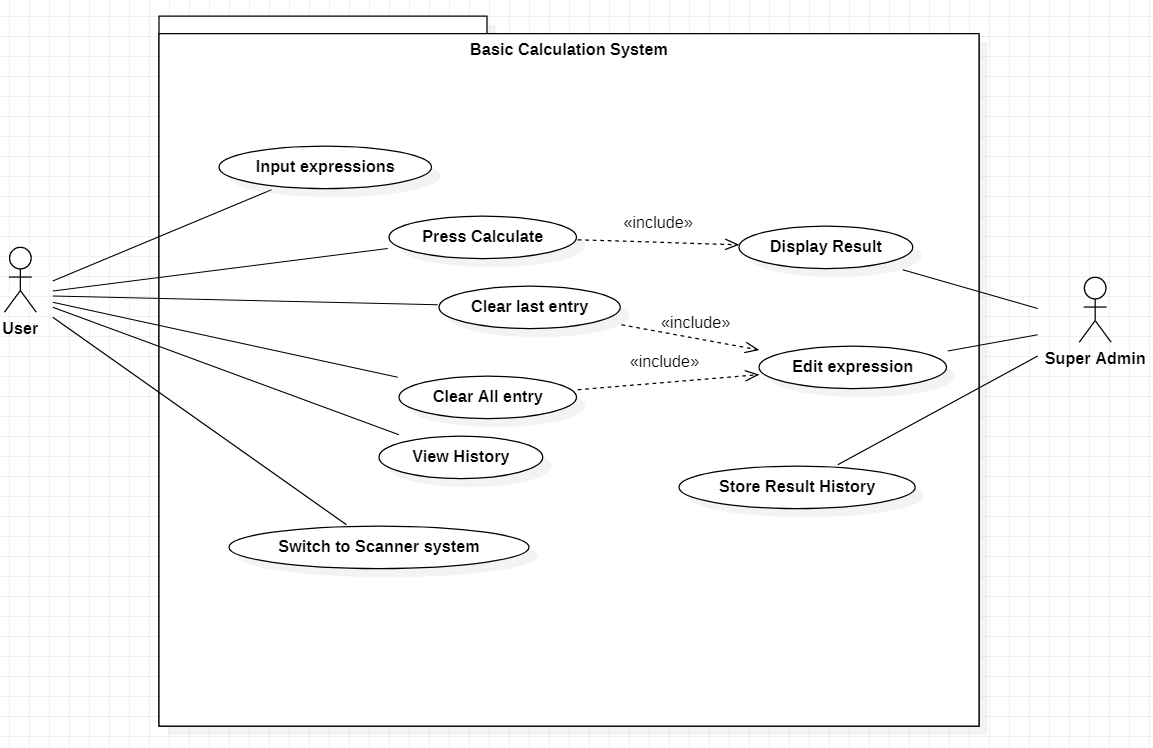


Figure 1: Use Case Diagram for Basic Calculation System

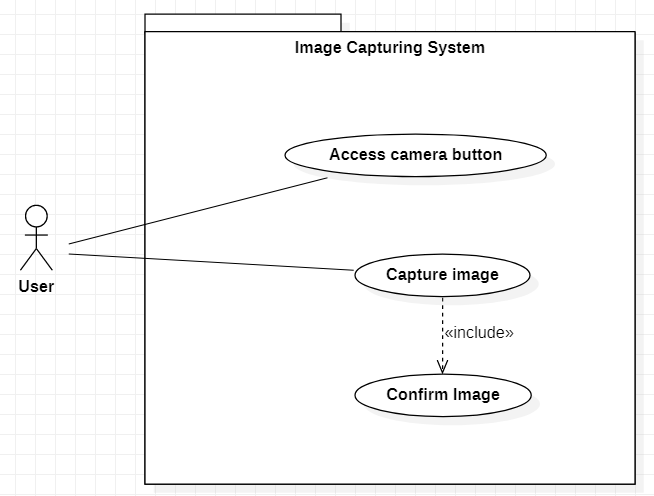
* Use Case Diagram for Image Capturing System

Figure 2: Use Case Diagram for Image Capturing System

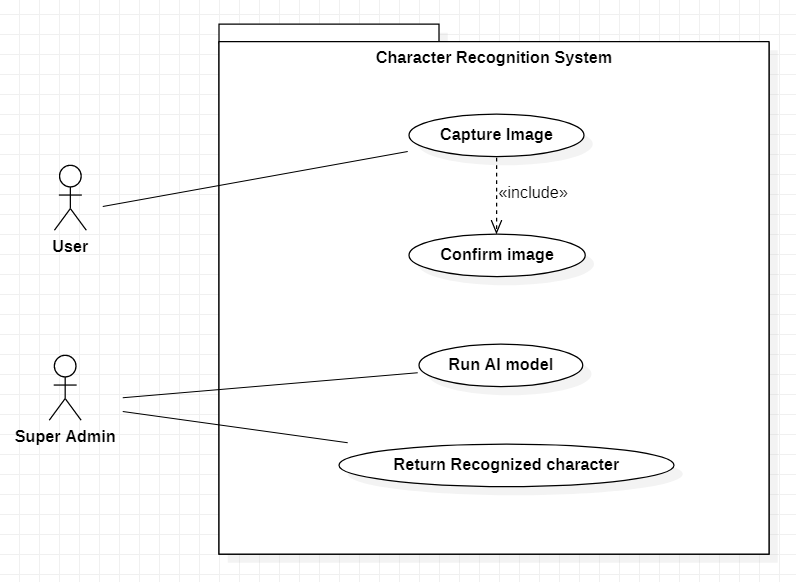
* Use Case Diagram for Character Recognition System

Figure 3: Use Case Diagram for Character Recognition System

### Structural Modeling

#### Class Diagram

Figure 4: Class Diagram

### Process Modeling

#### Context Diagram

* Context Diagram for Character Recognition System

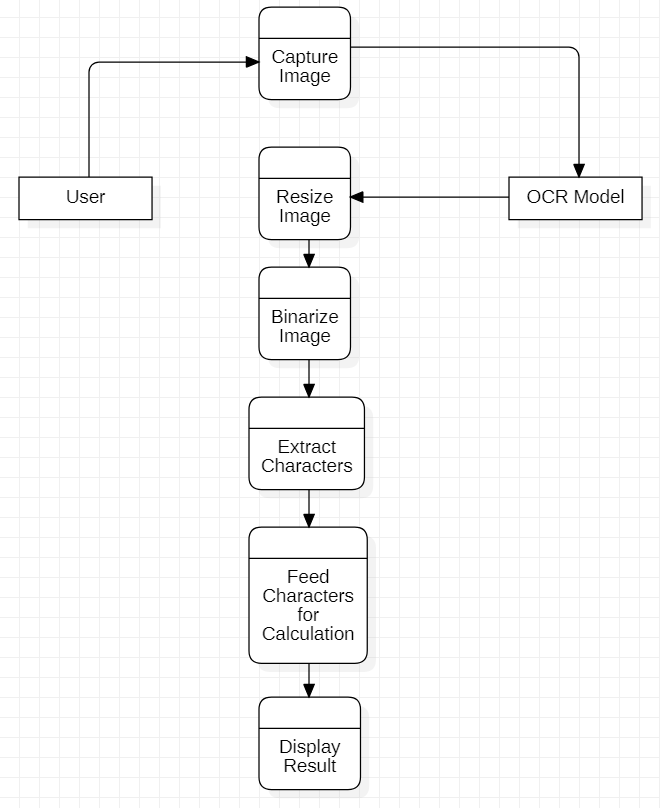


Figure 5: Context Diagram for Image Capturing System

* Context Diagram for Character Recognition System

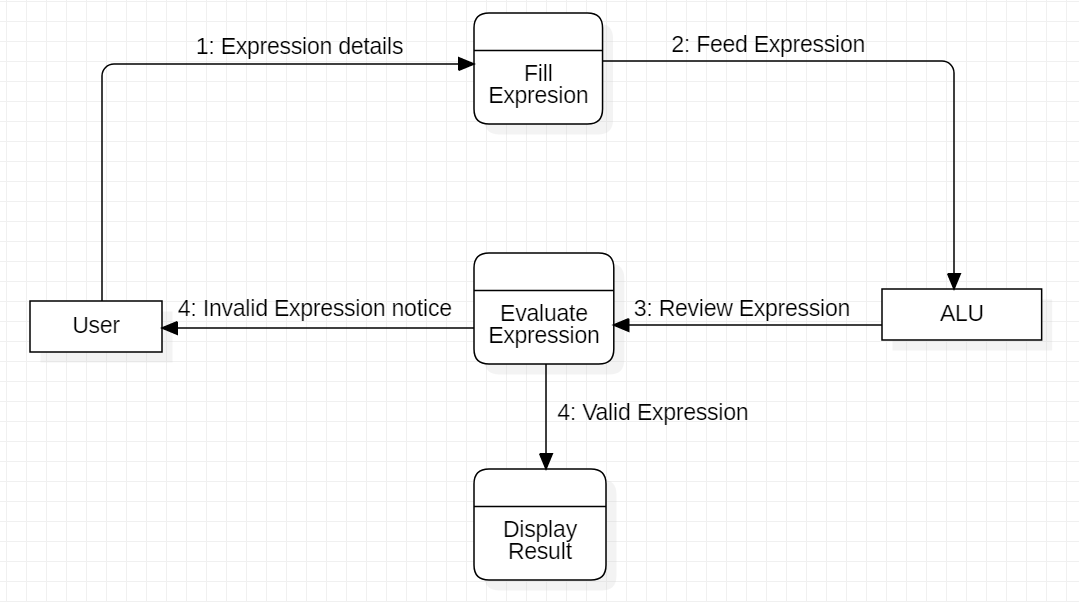


Figure 6: Context Diagram for Character Recognition System

### UI Model

#### Wireframe

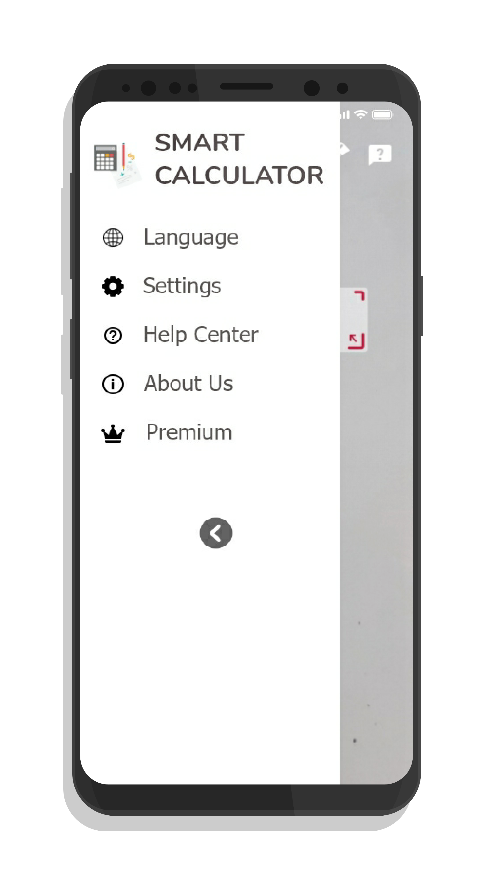
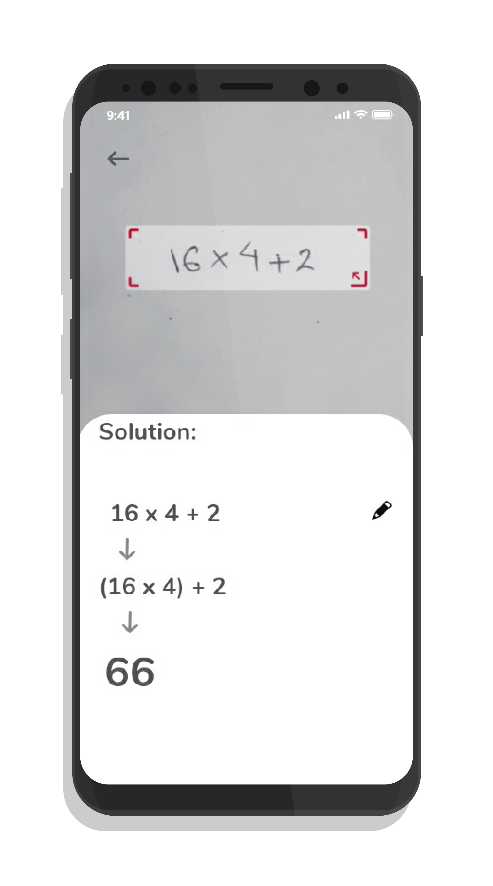
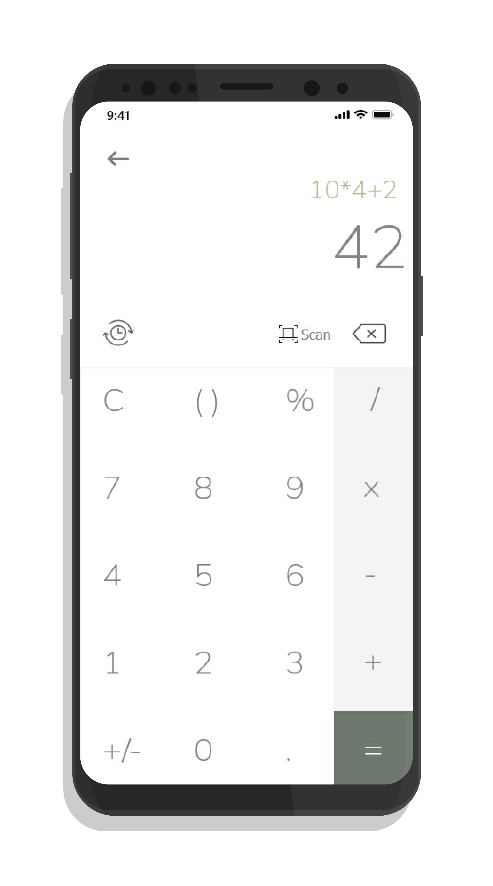
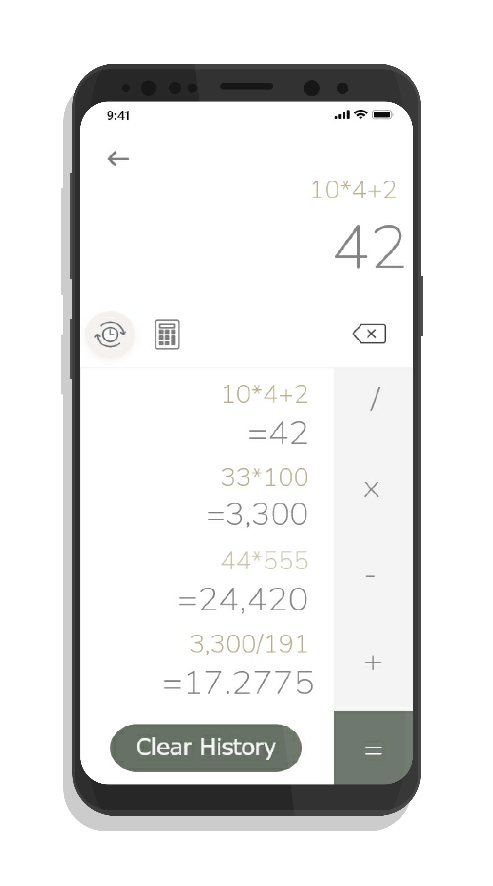


Figure 7: History Screen

Figure 8: Calculator Screen

Figure 9: Solution Screen

Figure 10: Navigation Screen

Figure 11: Main Screen

# Test Plan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **PRECONDITION** | **TEST SCENARIO DESCRIPTION** | | **TO TEST THE BASIC CALCULATION FUNCTIONALITY** | | | |  |
| **TS01** | User should have an android device with the app installed | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POSTCONDITION** | **ACTUAL RESULT** | **STATUS** |
| Doing basic arithmetic calculation | 1. Open the app 2. Navigate to calculator screen 3. Input arithmetic expression 4. Click evaluate button | 1\*2+3 | 5 | Both the input and result should be displayed on the screen |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **PRECONDITION** | **TEST SCENARIO DESCRIPTION** | | **TO TEST THE SCIENTIFIC CALCULATION FUNCTIONALITY** | | | |  |
| **TS02** | User should have an android device with the app installed | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POSTCONDITION** | **ACTUAL RESULT** | **STATUS** |
| Doing scientific arithmetic calculation | 1. Open the app 2. Navigate to calculator screen 3. Input scientific expression 4. Click evaluate button | Sin60+Cos30 | -0.15559 | Both the input and result should be displayed on the screen |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **PRECONDITION** | **TEST SCENARIO DESCRIPTION** | | **TO TEST THE HISTORY FUNCTIONALITY** | | | |  |
| **TS03** | User should have performed some calculations | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POSTCONDITION** | **ACTUAL RESULT** | **STATUS** |
| Checking the calculation history | 1. Open the app 2. Navigate to calculator screen 3. Click on history icon | - | - | All the previous calculation history should be displayed. |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **PRECONDITION** | **TEST SCENARIO DESCRIPTION** | | **TO TEST THE RECOGNITION OF MATHEMATICAL NOTATION** | | | |  |
| **TS04** | User should be in the calculator screen | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POSTCONDITION** | **ACTUAL RESULT** | **STATUS** |
| Testing the recognition of scientific notations. | 1. Open the app 2. Navigate to calculator screen 3. Input expression with mathematical notation 4. Click evaluate button | 5! | 120 | Both the input and result should be displayed on the screen |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **PRECONDITION** | **TEST SCENARIO DESCRIPTION** | | **TO TEST THE IMAGE CAPTURING FUNCTIONALITY** | | | |  |
| **TS05** | User should have navigated to camera screen | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POSTCONDITION** | **ACTUAL RESULT** | **STATUS** |
| Capturing image for calculation | 1. Open the app 2. Navigate to camera screen 3. Click on capture button 4. Confirm image | - | - | Characters should be extracted |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **PRECONDITION** | **TEST SCENARIO DESCRIPTION** | | **TO TEST THE RECOGNITON OF HANDWRITTEN CHARACTERS** | | | |  |
| **TS06** | User should have navigated to camera screen | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POSTCONDITION** | **ACTUAL RESULT** | **STATUS** |
| Testing recognition of handwritten characters | 1. Open the app 2. Navigate to camera screen 3. Click on capture button 4. Confirm image | - | - | Characters should be extracted |  |  |