

# **REQUIREMENT SPECIFICATIONS DOCUMENT**

FOR

**MR. CHEF**

**Your Kitchen Assistant**

Version 1.0

**Syndicate**

NC Muhammad Mohsin Zafar

NC Muhammad Sufyan

NC Muhammad Awab

**Military College of Signals**

**NUST**

**Project Supervisor**

Dr. Hammad Afzal

# Table of Contents

<b>Table of Figures.....</b>	<b>3</b>
<b>1. Introduction.....</b>	<b>4</b>
1.1 Purpose.....	4
1.2 Document Conventions.....	4
1.3 Intended Audience and Reading Suggestions.....	5
1.4 Definitions.....	6
1.5 Document Scope .....	7
1.6 References.....	7
<b>2. Overall Description.....</b>	<b>7</b>
2.1 Product Perspective.....	7
2.2 Product Functions .....	8
2.3 User Classes and Characteristics.....	8
2.4 Operating Environment.....	9
2.5 Design and Implementation Constraints .....	10
2.6 User Documentation .....	11
2.7 Assumptions and Dependencies.....	12
<b>3. External Interface Requirements .....</b>	<b>13</b>
3.1 User Interfaces .....	13
3.2 Hardware Interfaces .....	15
3.3 Software Interfaces .....	15
3.4 Communications Interfaces.....	16
<b>4. System Features .....</b>	<b>16</b>
4.1 Use Cases .....	17
4.2 User Registration .....	18
4.3 Selection from Available Recipes.....	19
4.4 Addition of New Recipes.....	21
4.5 Image Acquisition.....	23
4.6 Identification of Objects.....	24
4.7 Control of Robotic Arm(s).....	25
4.8 Acquisition of Feedback from Sensors .....	27
4.9 Progress Status of Cooking.....	28
4.10 Homing .....	29
<b>5. Other Nonfunctional Requirements.....</b>	<b>30</b>
5.1 Availability .....	30
5.2 Reliability.....	30
5.3 Maintainability .....	30
5.4 Security .....	31
5.5 Environmental.....	31
5.6 Legal .....	31
5.7 Software Quality Attributes .....	31
<b>Bibliography .....</b>	<b>32</b>

## Revision History

Name	Date	Change(s) Made	Version
0.9	13 <sup>th</sup> Nov, 2017	Use Case Diagram Added	1.0

## **Table of Figures**

<b>Figure 1 - Component Diagram .....</b>	<b>8</b>
<b>Figure 2 - Context Diagram .....</b>	<b>17</b>
<b>Figure 3 - Android Application Use Cases.....</b>	<b>17</b>

# **1. Introduction**

The section “Introduction” of the Software Requirements Specification; which from here on will be called “SRS”, provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the document. The aim of this document is to present detailed description of the project “MR. CHEF” in which requirements have been laid down for development of basic prototype. The detailed requirements of, “MR. CHEF” are provided in this document.

## **1.1 Purpose**

This document covers the software requirement specifications for project “MR. CHEF”. This document is meant to outline the features and requirements of “MR. CHEF”, to serve as a guide to the developers on one hand and a software validation document for the prospective client/stakeholders on the other.

## **1.2 Document Conventions**

This section describes the standards followed while writing this document.

### **1.2.1 Headings**

Heading are prioritized in a numbered fashion, the highest priority heading having a single digit and subsequent headings having more numbers, per their level.

All the main headings are titled as follows: single digit number followed by a dot and the name of the section (All bold Times New Roman, size 18, Left Aligned).

All second level sub headings for every sub section have the same number as their respective main heading, followed by one dot and subsequent sub heading number followed by name of the sub section (All bold Times New Roman, size 16).

Further sub headings, i.e. level three and below, follow the same rules as above for numbering and naming, but different for font (All bold Times New Roman, size 14).

### **1.2.2 Figures**

All figures in this document have captions, and are numbered. Context and flow diagrams are based on UML standards.

### **1.2.3 Reference**

All references in this document are provided where necessary, however where not present, the meaning is self-explanatory. All ambiguous terms have been clarified in the glossary at the end of this document.

### **1.2.4 Links to web pages**

All links have been provided with underlined font, the title of the web page or e-book is written at the top of the link and the title may be searched on google to pinpoint to the exact address.

### **1.2.5 Basic Text**

All other basic text appears in regular, size 12 Times New Roman. Every paragraph explains one type of idea.

## **1.3 Intended Audience and Reading Suggestions**

The intended audience for this SRS Document include the project supervisor, “MR. CHEF” syndicate, BE CSE 20, UG project evaluation team, and other stakeholders at CSE Department, MCS.

### **1.3.1 Project Supervisor**

It will help the supervisor to supervise the project and guide the team in a better way. This document will be used by respected supervisor to check whether all the requirements have been understood and in the end whether the requirements have been implemented properly or not.

### **1.3.2 Project Syndicate (developers, testers, and documentation writers)**

For FYP group members, this document will provide the guideline for developing and testing the project.

### **1.3.3 UG Project Evaluation Team:**

It will help the evaluation team to evaluate the progress of FYP. The document will provide the evaluators with the scope, requirements and details of the project to be built. It will also be used as basis for the evaluation of the implementation and final project.

### **1.3.4 Reading suggestions**

The SRS begins with the title and table of contents. All level 1 and level 2 headings are given in the table of contents, but the lower sub headings are not included. Each main heading is succeeded by several sub headings, which are all in bold format. The document overview is given at the start, succeeded by the complete detailed features, including both functional and non-functional requirements described in section 4 & section 5 of the document. The entire interfaces are also described. The SRS ends with appendices, including a glossary.

## **1.4 Definitions**

user. The person who operate or interact directly with the product.

raspberry pi. A credit card sized computer.

arduino. It is a physical programmable circuit board.

payload weight. Maximum weight that a robotic arm can lift/handle.

critical temperature. Maximum temperature that system can bear.

jumper wires. It is an electrical wire or group of them in a cable with a cable or connector at each end used to interconnect different components without the need of soldering.

objects. In this document the word object is mainly used to represent crockery, utensils and ingredients.

## 1.5 Document Scope

The document includes detailed description of project. Functional and Non-Functional Requirements have been laid down to lay foundation for development team for developing the project and testing team to test and verify the developed project.

## 1.6 References

### 1.6.1 IEEE Computer Society Conventions

- **Use Case Modeling Guidelines**, which documents the guidelines used to develop the use case model specifying the functional requirements in this specification.  
[http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=787548](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=787548)
- **System Requirements Specification Content and Format Standard**, which specifies the content and format of this specification.  
[http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?tp=&isnumber=15571&arnumber=720574&punumber=5841](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=15571&arnumber=720574&punumber=5841)
- **System Requirements Specification Template**, which provides the skeleton for this specification.  
[http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?tp=&isnumber=16016&arnumber=741940&punumber=5982](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=16016&arnumber=741940&punumber=5982)

## 2. Overall Description

### 2.1 Product Perspective

The idea of the project is to realize implementation of robotic arm(s) that shall carry out activities related to cooking in kitchen, approximately same as a human would do with task initiation done by the user thus providing a semi-autonomous system to automate kitchen tasks as much as possible. The goal is to take home automation a step further

## 2.2 Product Functions

The key features of “MR. CHEF” are highlighted below:

1. An Android Application providing an interface to user for interacting with robotic system
2. Connection of Android Application to Raspberry Pi over the internet
3. Object Identification by Raspberry Pi using a Camera
4. Control of Robotic Arms’ Movements by Raspberry Pi via Arduino
5. Use of Camera by Raspberry Pi after fixed intervals to verify status of meal being cooked.

Below is the component diagram for the project;

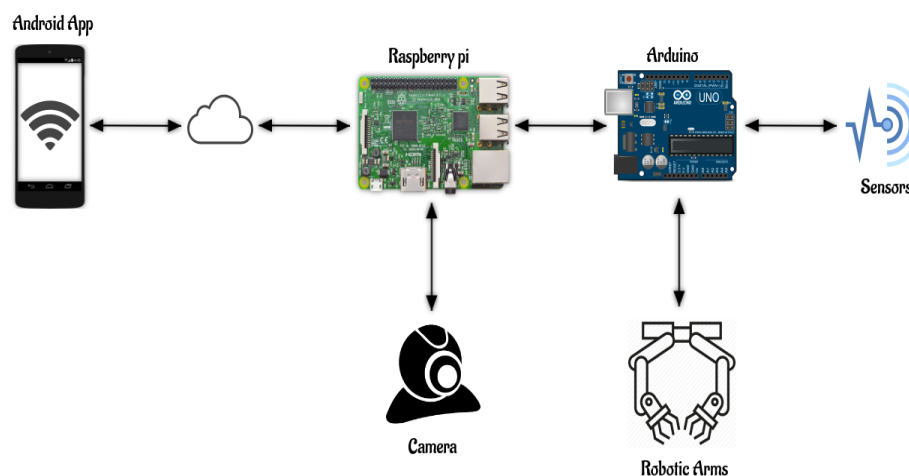


Figure 1 - Component Diagram

## 2.3 User Classes and Characteristics

### 2.3.1 Summary of User Classes

The following section describes the types of users of the “MR. CHEF”. There are explanations of the user followed by the interactions the user(s) shall be able to make with the software.



### **2.3.1.1 User who wants to add a new Recipe**

The user in this class is basically one who intends on adding a new recipe to the system by interacting with Android Application. User will have to follow a very specifically designed form to add a new recipe. Upon successful entry, user shall be notified of success and vice versa in other case. The user in this class can be categorized as a regular user.

### **2.3.1.2 User who initiates cooking**

The user in this class is basically one who selects a specific recipe from available recipes being shown in Android Application and then initiates cooking of recipe by the robotic system, by giving command via Application. Android Application shall notify user of success after recipe has been cooked or failure in case otherwise. The user in this class can be categorized as a regular user.

### **2.3.1.3 Testers/Development Team**

The user in this class is basically one who will be interacting with the system during development and testing phases to verify that whether product being developed is in accordance with SRS. The user in this class can be categorized as a regular user.

## **2.4 Operating Environment**

The sub-sections below give a brief description of environment, hardware & software based requirements for operation of “MR. CHEF”.

### **2.4.1 Environment**

The system shall operate in a known environment. Customized workstation shall be designed to facilitate robotic system to perform tasks easily.

### **2.4.2 Hardware**

“MR. CHEF” shall operate, either directly or indirectly, with the following external hardware:

- **Camera Mounted at the Work Station:** The camera shall be used to obtain real time pictures of the workstation that shall further help in identification of objects and status of meal being prepared.
- **Cable:** The data (pictures/snaps) shall be transferred from camera using cable to the raspberry pi.
- **Raspberry Pi:** It shall act as mini-computer with all necessary software installed on it. It shall receive images from the camera. Also, Raspberry Pi shall receive command of 'initiation' from the Android App. It shall also be responsible to control Arduino
- **Arduino:** This being the micro-controller shall be responsible for controlling robotic arm(s).
- **Robotic Arm(s):** Two 6-Axis Robotic arms shall be used to perform tasks of kitchen.
- **Sensor(s):** Sensors such as; Proximity Sensor, Temperature Sensor, Ultrasonic Sensor shall be used to achieve accuracy and safety while performing different tasks.

### **2.4.3 Software**

- Linux: Raspbian Jessie
- OpenCV 3
- IDE: Python IDLE / PyCharm
- IDE: Android Studio

## **2.5 Design and Implementation Constraints**

- Internet Connection between Android Application and Raspberry Pi is necessary for robotic system to receive commands and send feedback accordingly.
- Robotic System shall only work in a completely known environment.
- Robotic arm(s) being only a prototype have limitation of payload weight approximately 400-500 grams.

- There is limitation of handling high temperatures and keeping in view safety measurements the system shall stop working if a critical temperature is detected.
- If any object is dislocated from its original position system shall not be able to detect change.
- System shall use induction stove for cooking purposes.
- Only those utensils shall be used for cooking that are compatible with induction stove.
- User shall only be able to enter/add new recipe in a specific format by only using available keywords.
- Since, our product is only a prototype and it can't undergo lot of testing therefore, "Add New Recipe" feature may not produce desirous results.

## **2.6 User Documentation**

A user manual will be provided to the users in which separate instructions will be given according to the user i.e. Regular user and the admin, developers and testers. It will include the details of the system's working. Help documents will also be a part of the system.

The project report will also be available for the users which will highlight the system features, working and procedures.

## **2.7 Assumptions and Dependencies**

Following dependencies and assumptions are being observed for the project;

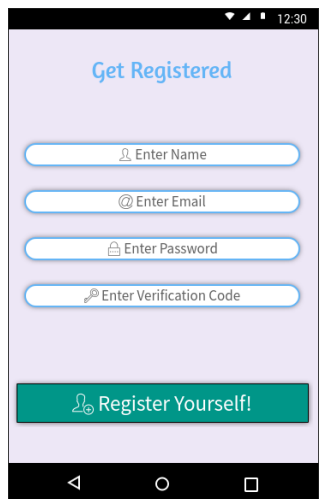
- There is dependency of environment in which our system shall work which means that the environment must be completely known to system else our system can't perform.
- Since, the project shall use prototype robotic arm(s) therefore, the project is completely dependent on the working performance of hardware components which also include handling weights of different ranges, handling high temperature etc.
- It is assumed that continuous electric power supply is ensured, the system shall not recover in case of power failure.
- System is also network dependent for initiating command, fetching data for database and receiving feedback from raspberry pi.
- Identification of objects is dependent on the quality resolution of camera, also the snapshots are expected to contain some noise.
- The system shall be given a Unique Identification Number for verification & registration of a user.

## 3. External Interface Requirements

### 3.1 User Interfaces

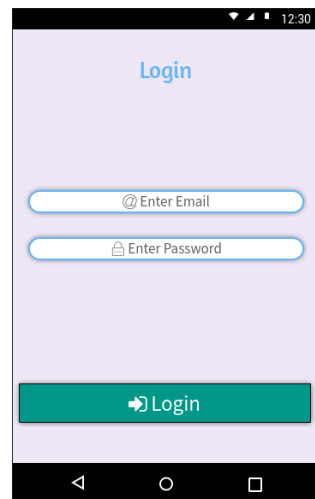
Below are the user interfaces for Android Application;

#### 3.1.1 Registration Screen



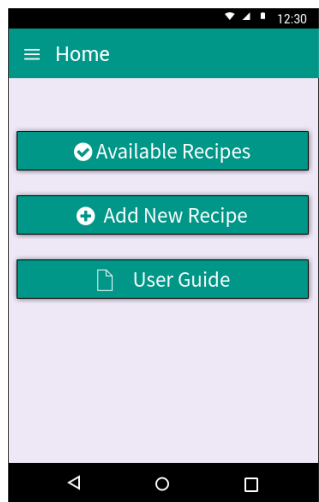
The Registration Screen (Get Registered) features a light purple background. At the top, the title "Get Registered" is displayed in blue. Below the title are four input fields, each with a blue border and a small icon on the left: "Enter Name" (person icon), "Enter Email" (email icon), "Enter Password" (lock icon), and "Enter Verification Code" (phone icon). At the bottom, there is a green button with a white person icon and the text "Register Yourself!". The screen is framed by a black status bar at the top showing the time 12:30 and a black navigation bar at the bottom with standard Android icons.

#### 3.1.2 Login Screen

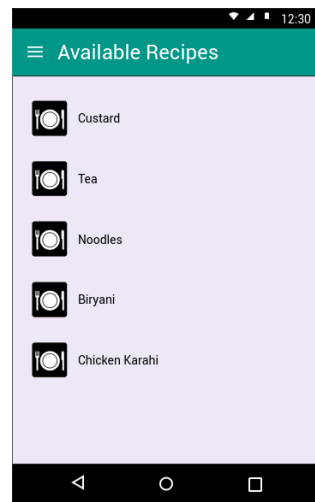


The Login Screen features a light purple background. At the top, the title "Login" is displayed in blue. Below the title are two input fields, each with a blue border and a small icon on the left: "Enter Email" (email icon) and "Enter Password" (lock icon). At the bottom, there is a green button with a white right-pointing arrow icon and the text "Login". The screen is framed by a black status bar at the top showing the time 12:30 and a black navigation bar at the bottom with standard Android icons.

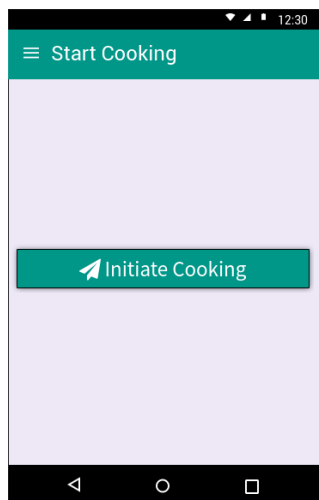
### 3.1.3 Home Screen



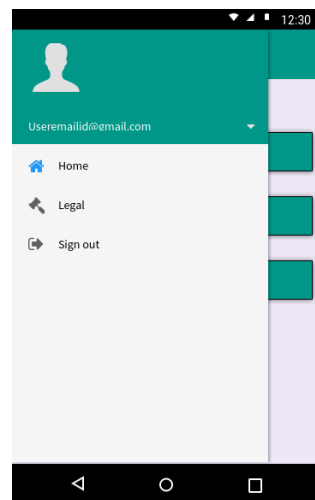
### 3.1.4 Available Recipes Screen



### 3.1.5 Initiate Cooking Screen



### 3.1.6 Drawer Activity



### 3.1.7 Add Recipe Screen

12:30

≡ Add New Recipe

Select Ingredients

☒ Ingredient 1 Qty 5

☒ Ingredient 2 Qty 4

☒ Ingredient 3 Qty 3

Select Utensils

☒ Utensil 1

☒ Utensil 2

☒ Utensil 3

Steps

Step 1: \_\_\_\_\_

Step 2: \_\_\_\_\_

Step 3: \_\_\_\_\_

+ Add Recipe

### 3.1.8 Splash Screen



## 3.2 Hardware Interfaces

- Image input shall be taken from camera in real time for identification of objects.
- Communication between camera and raspberry pi shall be done using a cable.
- Raspberry Pi shall communicate with Arduino via serial communication wire.
- Arduino shall control Servo Motors of Robotic Arm(s)
- Arduino shall receive feedback from different sensors and shall communicate it to Raspberry Pi

## 3.3 Software Interfaces

- Raspberry Pi shall be used as a mini computer to run code which shall control whole robotic system.
- OpenCV shall be used to process images.
- Android Application shall be used to provide an interface to user for interacting with robotic system.
- Database

### **3.4 Communications Interfaces**

- Android Application shall communicate with Raspberry Pi over the internet.
- Raspberry Pi shall communicate with Arduino via Serial Communication Wire.
- Raspberry Pi shall receive feed from camera via Cable.
- Arduino shall control Robotic Arm(s) via Jumper Wires.
- Arduino shall receive feedback from sensors via Jumper Wires.

## **4. System Features**

This section illustrates organization of functional requirements for the project “MR. CHEF” by system features:

- i. User Registration
- ii. Selection from Available Recipes
- iii. Addition of New Recipes
- iv. Image Acquisition
- v. Identification of Objects
- vi. Control of Robotic Arms(s)
- vii. Feedback Acquisition from sensors
- viii. Progress Status of Cooking
- ix. Homing



Following is the Context Diagram;

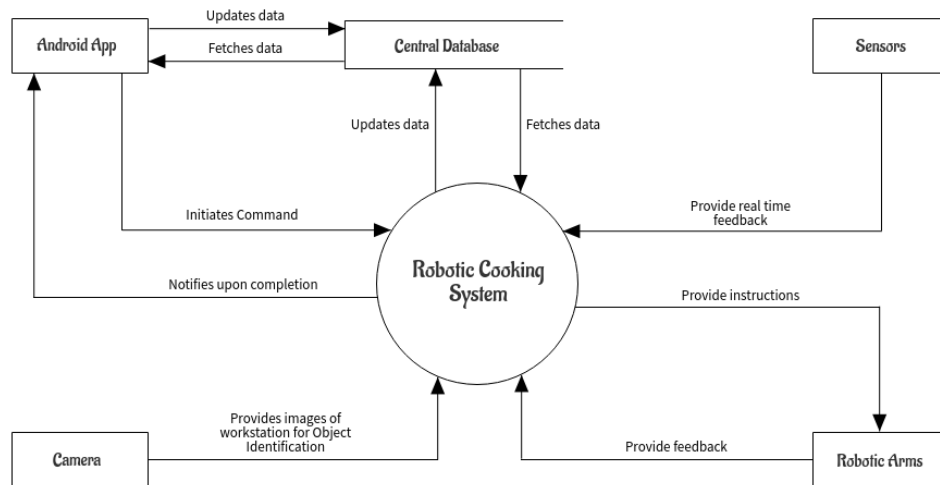


Figure 2 - Context Diagram

## 4.1 Use Cases

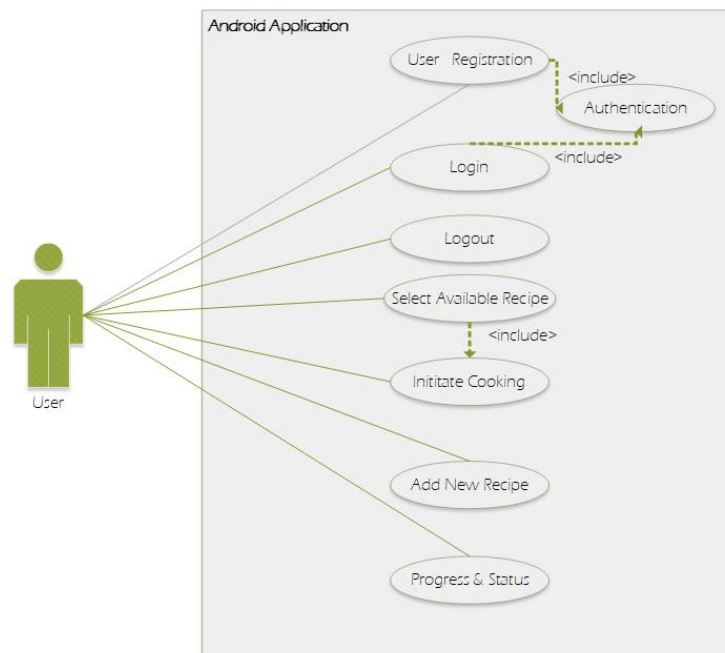


Figure 3 - Android Application Use Cases

Based on above presented Use Case Diagram, several functional requirements have been extracted and are discussed in Sections 4.2, 4.3 & 4.4 with Requirements ranging from, REQ-1 to REQ-19

## 4.2 User Registration

### 4.2.1 Description

This feature enables the user to get himself/herself register with the system and more specifically the Android Application. Also, this feature will ensure authenticity of the user.

### 4.2.2 Stimulus/Response Sequences

<b>4.2.2.1 Normal Path: User Registers with Application successfully</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>User must open the provided Android Application.</li> </ul>
<b>Interactions</b> <ul style="list-style-type: none"> <li>The user clicks “Register” button.</li> </ul>
<b>Post conditions</b> <ul style="list-style-type: none"> <li>“Registration Successful” message is displayed.</li> <li>Afterwards, user shall be taken to Home Screen of the Application.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li><b>Criticality:</b> High</li> <li><b>Probability of Defects:</b> Low</li> <li><b>Risk:</b> Medium</li> </ul>

<b>4.2.2.2 Exceptional Path: Error message is displayed</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>Application is not connected to Internet.</li> </ul>

<ul style="list-style-type: none"> <li>• User enters invalid credentials.</li> <li>• User enters invalid verification code.</li> </ul>
<b>Interactions</b> An exception is thrown in the system.
<b>Post conditions</b> <ul style="list-style-type: none"> <li>• Error message is displayed.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li>• <b>Criticality:</b> High</li> <li>• <b>Probability of Defects:</b> Low</li> <li>• <b>Risk:</b> High</li> </ul>

### 4.2.3 Functional Requirement

**REQ-1** The Application shall enable user to get himself/herself register with the System.

**REQ-2** The Application shall take User's credentials including Name, Email address & a Password, and a Unique Identification Number for the product; as input.

**REQ-3** The Application shall authenticate user's input data by matching verification code with System's database.

**REQ-4** Only, in case of successful Authentication, the application shall take user to the Home Screen.

**REQ-5** In case, user signs out of the application, user shall only be able to use Application's functionality once user has signed back into the application after providing email address & password, as input to the application.

## 4.3 Selection from Available Recipes

### 4.3.1 Description

This feature enables the user to select a specific recipe from a list of available recipes and shall allow user to give command to system to start cooking the recipe.

### 4.3.2 Stimulus/Response Sequences

<b>4.3.2.1 Normal Path: User selects Recipe &amp; initiates cooking</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>User must be signed in to the Application.</li> <li>User clicks on the button, “Select Recipe”.</li> </ul>
<b>Interactions</b> <ul style="list-style-type: none"> <li>User selects a recipe from list of available recipes.</li> </ul>
<b>Post conditions</b> <ul style="list-style-type: none"> <li>User is provided with an option of “Initiate Cooking”.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li><b>Criticality:</b> High</li> <li><b>Probability of Defects:</b> Low</li> <li><b>Risk:</b> Low</li> </ul>

<b>4.3.2.2 Exceptional Path: Selection Failed/Cooking Initiation Failed</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>The List of Available Recipes failed to Load.</li> <li>Internet Connection Lost after or while user clicks on “Initiate Cooking” Button.</li> </ul>
<b>Interactions</b> <p>An error/failure exception message is raised by the Application.</p>
<b>Post conditions</b> <ul style="list-style-type: none"> <li>User is displayed with message to refresh list.</li> <li>User is displayed with message to connect to Internet.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li><b>Criticality:</b> High</li> <li><b>Probability of Defects:</b> Low</li> </ul>

- **Risk:** Medium

### 4.3.3 Functional Requirement

**REQ-6** The Application shall present user with a List of All Available Recipes if and when user clicks on the button of “Available Recipes” while at Home Screen.

**REQ-7** The Application shall allow user to select only one Recipe from the list.

**REQ-8** The Application shall enable/show a button, “Initiate Cooking” upon successful selection of Recipe by the user.

**REQ-9** The Application shall send command to Raspberry Pi via Internet to Initiate the Cooking of Selected Recipe, when user clicks on “Initiate Cooking” button,

**REQ-10** After the initiate cooking command has been send to Raspberry Pi, the Application shall take user to the Progress and Status Screen.

## 4.4 Addition of New Recipes

### 4.4.1 Description

This feature enables the user to add new recipes to the Application.

### 4.4.2 Stimulus/Response Sequences

<b>4.4.2.1 Normal Path: New Recipe Added Successfully</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>• The user is signed in to the Application.</li> <li>• The user is at Home Screen.</li> </ul>
<b>Interactions</b> <ul style="list-style-type: none"> <li>• The user clicks on the button “Add New Recipe”.</li> </ul>
<b>Post conditions</b> <ul style="list-style-type: none"> <li>• User fills forms for Adding New Recipe.</li> </ul>

<ul style="list-style-type: none"> <li>• New Recipe is added successfully.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li>• <b>Criticality:</b> High</li> <li>• <b>Probability of Defects:</b> High</li> <li>• <b>Risk:</b> High</li> </ul>

<b>4.4.2.2 Exceptional Path: Addition of New Recipe failed</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>• User enters invalid data.</li> <li>• Local database is not responsive.</li> </ul>
<b>Interactions</b> An exception is thrown by the Application.
<b>Post conditions</b> <ul style="list-style-type: none"> <li>• “Add New Recipe: Failed” message is displayed.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li>• <b>Criticality:</b> High</li> <li>• <b>Probability of Defects:</b> High</li> <li>• <b>Risk:</b> High</li> </ul>

### 4.4.3 Functional Requirement

**REQ-11** The Application shall take user to “Add New Recipe” Screen, when user clicks on “Add New Recipe” button while at Home Screen.

**REQ-12** The application shall ask user for a unique Recipe Name.

**REQ-13** The Application shall ask user to Add Ingredients and their quantity.

**REQ-14** The Application shall allow user to select ingredients only that are already listed in the system.

**REQ-15** The Application shall ask user to select utensils and crockery from list of available utensils and crockery.

**REQ-16** The Application shall ask user to add steps for recipe, using only the keywords provided in Application.

**REQ-17** The Application shall check for validity of added steps in REQ-16.

**REQ-18** The Application shall prompt user to “Add” or “Cancel”, in case of successful validation of entered steps.

**REQ-19** The Application shall add newly entered recipe to list of available recipes.

## 4.5 Image Acquisition

### 4.5.1 Description

This feature enables the system to acquire images from camera mounted at workstation. The images will be fed into the system for further processing.

### 4.5.2 Stimulus/Response Sequences

<b>4.5.2.1 Normal Path: Images sent for processing</b>
<b>Preconditions</b> <ul style="list-style-type: none"><li>• Android Application must have already given command to Raspberry Pi for initiating cooking.</li><li>• The camera captures the images in real time.</li></ul>
<b>Interactions</b> <ul style="list-style-type: none"><li>• The captured images are sent to the system for processing.</li></ul>
<b>Post conditions</b> <ul style="list-style-type: none"><li>• Captured images are fed to object identification algorithms.</li></ul>
<b>Categorization</b> <ul style="list-style-type: none"><li>• <b>Criticality:</b> High</li><li>• <b>Probability of Defects:</b> Medium</li><li>• <b>Risk:</b> High</li></ul>

<b>4.5.2.2 Exceptional Path: Warning LED starts blinking</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>The camera is disconnected/malfunctioned.</li> </ul>
<b>Interactions</b> An error signal is raised by the system
<b>Post conditions</b> <ul style="list-style-type: none"> <li>The warning LED attached with Raspberry Pi starts blinking.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li><b>Criticality:</b> High</li> <li><b>Probability of Defects:</b> Low</li> <li><b>Risk:</b> High</li> </ul>

### 4.5.3 Functional Requirement

**REQ-20** The system shall be able to acquire images of workstation in real time for further processing.

## 4.6 Identification of Objects

### 4.6.1 Description

This feature identifies objects placed at the workstation by processing the captured images.

### 4.6.2 Stimulus/Request Sequences

<b>4.6.2.1 Normal Path: Successfully identified objects</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>The images are sent for processing.</li> </ul>



<b>Post conditions</b>
<ul style="list-style-type: none"> <li>• Objects specific to “Recipe” under progress are identified by the system</li> </ul>
<b>Categorization</b>
<ul style="list-style-type: none"> <li>• <b>Criticality:</b> High</li> <li>• <b>Probability of Defects:</b> Medium</li> <li>• <b>Risk:</b> Medium</li> </ul>

### 4.6.3 Functional Requirement

**REQ-21** The system shall be able to identify objects that are placed on the workstation where objects include; ingredients, cutlery, utensils & stove.

**REQ-22** The system shall be able to distinguish objects that are specific to Recipe under progress, from all identified objects.

## 4.7 Control of Robotic Arm(s)

### 4.7.1 Description

Once the system has identified all objects that are to be used in cooking procedure to follow, the control shall then be transferred to Robotic Arm(s) that will be responsible for carrying out task/steps as per the recipe.

### 4.7.2 Stimulus/Response Sequence

<b>4.7.2.1 Normal Path: Robotic Arm(s) Complete All Steps of Recipe Successfully</b>
<b>Preconditions</b>
<ul style="list-style-type: none"> <li>• All objects that are to be used in cooking are identified.</li> <li>• System is aware of the coordinates of all objects that are to be used.</li> </ul>
<b>Interactions</b>
<ul style="list-style-type: none"> <li>• System shall control robotic arm(s) to complete all steps of recipe to be cooked.</li> </ul>
<b>Post conditions</b>

<ul style="list-style-type: none"> <li>• All steps of recipe to be cooked are completed.</li> <li>• System shall notify Android Application that Recipe has been cooked.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li>• <b>Criticality:</b> High</li> <li>• <b>Probability of Defects:</b> High</li> <li>• <b>Risk:</b> High</li> </ul>

<b>4.7.2.2 Exceptional Path: Robotic Arm(s) malfunctioned or couldn't complete all steps</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>• Hardware Failure.</li> <li>• Power Failure.</li> <li>• Connection to Robotic Arm(s) Lost.</li> </ul>
<b>Interactions</b> System halts and notifies Android Application of the Failure.
<b>Post conditions</b> <ul style="list-style-type: none"> <li>• Android Application notifies user of the failure by displaying a Failure Message.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li>• <b>Criticality:</b> High</li> <li>• <b>Probability of Defects:</b> Low</li> <li>• <b>Risk:</b> High</li> </ul>

### 4.7.3 Functional Requirements

**REQ-23** The system shall be able to effectively control Robotic Arm(s) to perform tasks as per the recipe.

**REQ-24** The system shall notify user via Android Application after successful completion of all task as per the recipe.

## 4.8 Acquisition of Feedback from Sensors

### 4.8.1 Description

This feature enables system to acquire feedback from different sensors including, temperature sensor, proximity sensor & ultrasonic sensor; while cooking is under progress.

### 4.8.2 Stimulus/Response Sequences

<b>4.8.2.1 Normal Path: Continuous feedback from sensors is obtained</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>Command to initiate cooking has been received by the system.</li> </ul>
<b>Interactions</b> <ul style="list-style-type: none"> <li>Feedback from sensors is obtained.</li> </ul>
<b>Post conditions</b> <ul style="list-style-type: none"> <li>Based on feedback from Temperature Sensor, the system shall see if arm(s) are approaching critical temperature.</li> <li>Feedback from Proximity Sensor, shall help system to know if robotic arm(s) are near the objects.</li> <li>Feedback from Motion Sensor, shall help system to control robotic arm(s) movements.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li><b>Criticality:</b> High</li> <li><b>Probability of Defects:</b> Medium</li> <li><b>Risk:</b> High</li> </ul>

<b>4.8.2.2 Exceptional Path: Sensor(s) unresponsive/malfunctioned</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>Hardware Failure.</li> <li>Power Failure.</li> </ul>

<ul style="list-style-type: none"> <li>• Connection to Sensor(s) Lost.</li> </ul>
<b>Interactions</b> System halts and notifies Android Application of the Failure.
<b>Post conditions</b> <ul style="list-style-type: none"> <li>• Android Application notifies user of the failure by displaying a Failure Message.</li> </ul>
<b>Categorization</b> <ul style="list-style-type: none"> <li>• <b>Criticality:</b> High</li> <li>• <b>Probability of Defects:</b> Low</li> <li>• <b>Risk:</b> High</li> </ul>

### 4.8.3 Functional Requirement

**REQ-25** The system shall be able to acquire feedback from Proximity Sensor, Ultrasonic Sensor & Temperature sensor to ensure effective continuation of tasks by Robotic Arm(s).

## 4.9 Progress Status of Cooking

### 4.9.1 Description

This feature enable system to keep track of progress of cooking.

### 4.9.2 Stimulus/Response Sequences

<b>4.9.2.1 Normal Path: Successful Completion of Recipe</b>
<b>Preconditions</b> <ul style="list-style-type: none"> <li>• Command for initiate cooking has been received by system.</li> <li>• Tasks that include usage of stove have begun.</li> </ul>
<b>Interactions</b> <ul style="list-style-type: none"> <li>• System shall use camera to capture images of pan on stove after regular intervals.</li> </ul>
<b>Post conditions</b>

- Using captured snaps and after processing them system shall determine whether task has been completed.

#### **Categorization**

- **Criticality:** High
- **Probability of Defects:** Medium
- **Risk:** High

### **4.9.3 Functional Requirement**

**REQ-26** The system shall take snaps of Cooking Pan while it has been placed on Stove after regular intervals (ranging from 15-30 seconds) to determine whether cooking task has completed or achieved desirous state before moving to next step or notifying application of completion.

## **4.10 Homing**

### **4.10.1 Description**

This feature enables robot to return to its original state after each step of recipe is completed.

### **4.10.2 Stimulus/Response Sequences**

#### **4.10.2.1 Normal Path: Robot returns to its original state**

##### **Preconditions**

- Command for initiate cooking has been received by system.
- A Step of recipe under progress has been completed.

##### **Interactions**

- System shall give command to robot to return to its original/initial state.

##### **Post conditions**

- Robot returns to its original/initial state.

**Categorization**

- **Criticality:** High
- **Probability of Defects:** Medium
- **Risk:** High

### 4.10.3 Functional Requirement

**REQ-27** The system shall command robot arms(s) to return to original/initial state after every single step of the recipe completes.

## 5. Other Nonfunctional Requirements

### 5.1 Availability

**NFR-1** The system shall always be available to receive command from the user and to start cooking.

### 5.2 Reliability

**NFR-2** The Android Application shall always show latest and up to date List of Available Recipes.

### 5.3 Maintainability

**NFR-3** The system shall use standard sensors & electrical components available in market which may be replaced in case of component failure.

**NFR-4** User Interface shall be independent of the back-end code responsible for interacting with hardware components therefore allowing changes to be done in either of the codes without affecting the other.

## **5.4 Security**

**NFR-5** Android Application shall not allow user to use application or system features unless registered and signed in.

**NFR-6** The application shall not require user to sign into the application each time except for the first time unless user signs out.

## **5.5 Environmental**

**NFR-7** The system shall use induction stove therefore reducing environmental damage caused by gas stoves.

## **5.6 Legal**

**NFR-8** System shall adhere to public food and health safety standards.

## **5.7 Software Quality Attributes**

### **5.7.1 Usability**

The graphical user interface of system shall be designed with usability as the priority. The app will be presented and organized in a manner that is both visually appealing and easy for the user to navigate.

The user shall be able to use application with a maximum of three training sessions.

User guide shall be provided within the Android Application.

## **Bibliography**

### **Similar projects at MCS**

- Mr. PEN (An Artist)
- WIRELESSLY SYNCHRONIZED ROBOTIC ARM
- HAND GESTURE BASED ROBOTIC ARM

### **External Resources**

- Source 1: <http://www.bbc.com/news/science-environment-32282131>
- Source 2: <http://www.moley.com/>