# SRS (Software Requirement specification)

- PURPOSE

The purpose of the project is to provide farmers and customers a software solution that delivers a scalable, secure, and reliable system that helps the farmers to sell their goods directly to the customer at a reasonable price avoiding the dealers which in turn benefits both the farmer and the consumer. The following SRS document will outline the features of the "Buy from Farmer" and the requirements that the project will adhere to in developing the software for the organization.

- INTENDED AUDIENCE

This document is used by developers for understanding the requirements of the project; also, the intended users of this document are project managers for planning and scheduling, testers to generate test cases, document writers for preparation of user manual, and for other end-users/stakeholders to validate their requirements

- PROJECT SCOPE

Buy from Farmer is aimed at selling the farmer's products directly to the customers without any dealers in between farmers and customers so that both the farmers and customers get benefitted. This system maintains a database of the farmer's production left so the customers can pre-order the required items in advance if the items are available. The objective of the project is to help the farmers to sell their products at reasonable prices to the customers. This project helps farmers as a virtual dealer.

The goal of the project is to reduce the time and cost incurred in access to the database and design a database feasible to provide appropriate results to the queries of all the stakeholders.

- OPERATING ENVIRONMENT

| | |
|---|---|
| Programming Language | HTML, PHP |
| IDE | NetBeans |
| Database | MySQL |

- DESIGN AND IMPLEMENTATION CONSTRAINTS

Design Constraints:

Design mechanisms are limited to the capabilities of HTML commands and MySQL DB. The user interface must be viewable on a monitor with a 1024x768 resolution or larger.

Safety and Security Considerations:

Since personal information of farmers and customers will be contained and accessed. So, safety and security considerations need to be taken into account. Personal and sensitive information cannot be accessed by unauthorized users.

- SYSTEM FEATURES

The end-user having the software installed at his desktop should be able to login using the valid credentials. Post validation of the credentials entered; the system allows access into the database. Upon access, the user should be able to perform operations like insert, delete and update his production records into the database, and also generate reports. Provision should be provided for the user to query the database to obtain various reports.

| # | Title of System Feature | Description | Priority | Functional Requirements |
|---|---|---|---|---|
| 1 | Login | This is used to login, register, and maintain security by authenticating the users. | High 9 | Registration feature:<br>1. The first step is to register and every user needs to have access to login. Users get registered using variables- **name, username, mobile number, email id, password and category**.<br>2. If the user is registered with the same email id previously, then the user's account will |

| | | | | |
|---|---|---|---|---|
| | | | | not be created and will be prompted to create an account with different email id.<br>3. After registration, the software will be redirected to the login page.<br>4. Users need to go through the registration process again if the error is prompted.<br><br><u>Login feature</u>:<br><br>1. Should accept the variables **username, password, and category** to which the user belongs to.<br>2. A case insensitive comparison is done for a user name and a case sensitive comparison is done for a password.<br>3. If the correct user id and password along with proper category selection are supplied then, Main Menu should be displayed.<br>4. If an invalid user id or password is entered, then the system should display an error message "**Login Denied**" and should give a try again option. |
| 2 | Menu | This feature behaves as a user interface to display the features of the database and guides the user to access the database | High<br>8 | This is the main menu of the application and contains four sub categories:<br><br>1. The user profile details which will contain variables of users such as **name, rating, email id, mobile number and address**.<br>2. The user can **update** the details in profile whenever required. |

| | | | | |
|---|---|---|---|---|
| | | | | 3. The **MyCart** where the items are added and moved to final billing to the payment menu.<br>4. If the user is a farmer then the cart option is not available.<br>5. The **Digital market** platform where the items are searched and added to the cart.<br>6. The **Home** is used to redirect to the main menu.<br><br>Prompt the user to make choice with the 2 types of menus available:<br><br>1. **Product Maintenance menu:** Add, delete, update items in the product database by the farmers.<br>2. **Customer Maintenance menu:** Add or delete items to the cart with their respective quantities. |
| 3 | Product/Farmer Maintenance Menu | This feature adds and deletes products, updates the products name with the existing price, and helps in product details. | High<br>7 | **Add Item:**<br><br>1. **Product id** should be auto-generated by the system.<br>2. Accept the **Product name** not exceeding 15 characters and minimum of 3 characters and product price as well as the **product quantity** available.<br>3. The product name can contain only alphabets and product price and quantity should contain only numeric values with units.<br>4. Product names cannot be duplicated.<br><br>**Delete Item:**<br><br>1. If the product is not available or expired then the farmer can delete the item. |

| | | | | |
|---|---|---|---|---|
| | | | | 2. When the farmer finishes deleting the product, the control should return to the "Product Maintenance Menu Screen.<br><br>**Update Item:**<br><br>1. Accept the **Product id & new Product name**.<br>2. When an invalid product id is entered, display an error message, and ask the farmer to re-enter the product id.<br>3. Product names cannot be duplicated.<br>4. When the farmer finishes updating the product price and name, the control should return to the "Product Maintenance Menu Screen"<br><br>**View all Products:**<br><br>Display all the product names along with the **product id** in a tabular format |
| 4 | Customer maintenance menu | This feature helps customers to see various farmers' products and their details in their respective locations and add to their cart whatever they wish to buy. | Medium 7 | **Add item:**<br><br>1. The customer can add the required products into the cart from the digital market.<br>2. The products added are placed in the customers cart with variables **product id, name, product price and quantity.**<br>3. The added product is not stored in the database until the payment process is completed.<br><br>**Delete item:** |

| | | | | |
|---|---|---|---|---|
| | | | | 1. They customer can also delete the products in the cart.<br>2. The item will be removed without being stored to the database.<br><br>**Review:**<br><br>1. The customers can give reviews to the products based on their interest.<br>2. The customer needs to provide the address and confirm the address details before moving to payment details. |
| 5 | Payment details | This feature helps the user to confirm the order by selecting the payment method and providing the proper details. | Medium 7 | **Payment process:**<br><br>1. The customer after proceeding for buying their order will be directed to the transaction page where payment is made.<br>2. The payment will contain variables such as **name, mode of payment and address** to confirm customers order.<br>3. After confirmation the order details are stored in the database.<br><br>**Cancel Payment:**<br><br>1. If the customer wishes to cancel the order, the order details stored in the database will be changed to cancel status.<br>2. Also, the farmers and the customers will be shown that the order has been cancelled. |

- EXTERNAL INTERFACE REQUIREMENTS

User Interfaces:

The 'Buy from Farmer' should be a simple application used by its stakeholders with/without much technical knowledge.

• Login Interface: This interface prompts the current farmers and customers to login using their credentials and also allows the upcoming farmers and the new customers to register. Also, if the entered credentials are wrong, an error message is popped up to ensure proper security.

• Menu Interface**:** This interface displays the home page where the farmer and customer will have the option to access their profile, cart and the online/digital market. The profile section is editable and can be changed according to the user.

• Product Interface: The interface is for the Farmers which has options to add or delete the products available in their farm. It is so user friendly that the farmers can choose their own interested language and explore the portal with ease.

• Customer Interface: This interface is for the customer side where they can select their products after referring to the digital market and add it to their respective carts and proceed to place the orders.

• Payment Interface: Using this interface, the transactions can be made directly from the customer to the farmer and the order is further processed for confirmation and delivery. In this way there are direct benefits for both the consumers and the farmers.

Hardware Interfaces:

This software does not require any direct hardware interfaces.

Software Interfaces:

| | |
|---|---|
| Operating System | Windows 10 |
| Programming Language | HTML, PHP |
| IDE | Netbeans |
| Database | MySQL |

In this application the software interfaces are:

- Windows 10 operating system serves as a platform for the development and deployment of the College Management system software.

- NetBeans IDE version 7.2.1 which comes with MySql DB can be used

- JDK 1.6 can be used for front end development.

- JDBC (Java Database Connectivity) drivers serve as an interface to connect to the Derby DB from the front end.

Communication Interfaces:

This software is a standalone system hence does not require any communication interfaces.

- OTHER NONFUNCTIONAL REQUIREMENTS

| S.no | Type of Requirement | Details |
|---|---|---|
| 1 | Performance Requirements | The Product management system should be able to run on a standalone computer. As the database grows the intended performance with respect to report generation and accessing of data should not be affected. |
| 2 | Availability Requirements | This software is completely replacing the traditional system which is in place, hence the system must be able to provide accurate data to all the users with valid credentials. |
| 3 | Training Requirements | The GUI should be user-friendly. |