**Software Requirements Specification for: Kisaan Bandhu**

**Project Version 1.0**

**Prepared by**

**Meet Bhanushali - 1711005**

**Govinda Patel - 1711038**

**Parth Sheth - 1711055**

**Shailesh Upadhyay - 1711061**

**<Organization> K J Somaiya College Of Engineering , Vidyavihar , Mumbai.**

**<Date created> 18/10/2020**

**Table of Contents**

Contents

[**1.** **Introduction** 4](#_Toc54101846)

[**1.1.** **Purpose** 4](#_Toc54101847)

[**1.2.** **Product Scope** 4](#_Toc54101848)

[**1.3.** **References** 4](#_Toc54101849)

[**2.** **Overall Description** 5](#_Toc54101850)

[**2.1.** **Product Perspective** 5](#_Toc54101851)

[**2.2.** **Product Functions** 6](#_Toc54101852)

[**2.3.** **Operating Environment** 6](#_Toc54101853)

[**2.4.** **Design and Implementation Constraints** 7](#_Toc54101854)

[**2.5.** **User Documentation** 7](#_Toc54101855)

[**2.6.** **Assumptions and Dependencies** 7](#_Toc54101856)

[3. **External Interface Requirements** 8](#_Toc54101857)

[**3.1.** **User Interfaces** 8](#_Toc54101858)

[**3.2.** **Hardware Interfaces** 10](#_Toc54101859)

[**3.3.** **Software Interfaces** 10](#_Toc54101860)

[**3.4.** **Communications Interfaces** 10](#_Toc54101861)

[**4.** **System Features** 11](#_Toc54101862)

[**4.1.** **SF1** 11](#_Toc54101863)

[**4.2.** **SF2** 11](#_Toc54101864)

[**4.3.** **SF3** 11](#_Toc54101865)

[**4.4.** **SF4** 11](#_Toc54101866)

[**4.5.** **SF5** 11](#_Toc54101867)

[**4.6.** **SF6** 11](#_Toc54101868)

[**4.7.** **SF7** 12](#_Toc54101869)

[**4.8.** **SF8** 12](#_Toc54101870)

[5. **Other Nonfunctional Requirements** 13](#_Toc54101871)

[**5.1.** **Performance Requirements** 13](#_Toc54101872)

[**5.2.** **Safety Requirements** 13](#_Toc54101873)

[**5.3.** **Security Requirements** 13](#_Toc54101874)

[**5.4.** **Software Quality Attributes** 13](#_Toc54101875)

[**5.4.1.** **Reliability** 13](#_Toc54101876)

[**5.4.2.** **Maintainability** 13](#_Toc54101877)

[**5.4.3.** **Availability** 14](#_Toc54101878)

[**5.4.4.** **Portability** 14](#_Toc54101879)

1. **Introduction**
   1. **Purpose**

**Kisaan Bandhu** will enable the farmer to directly reach the buyers such as bulk buyers and supermarkets hence cutting out the extra cost due the increase in the cycle due to the profit margins of the Dealers. Also the entire profit made by selling the product is received by the farmer. This application will be mainly used by farmers after crop season who need some source of platform to directly reach the buyers by eliminating this system will also enable the farmers to directly receive some orders by large industries which will be contract based deals which will be managed by our application.

* 1. **Product Scope**

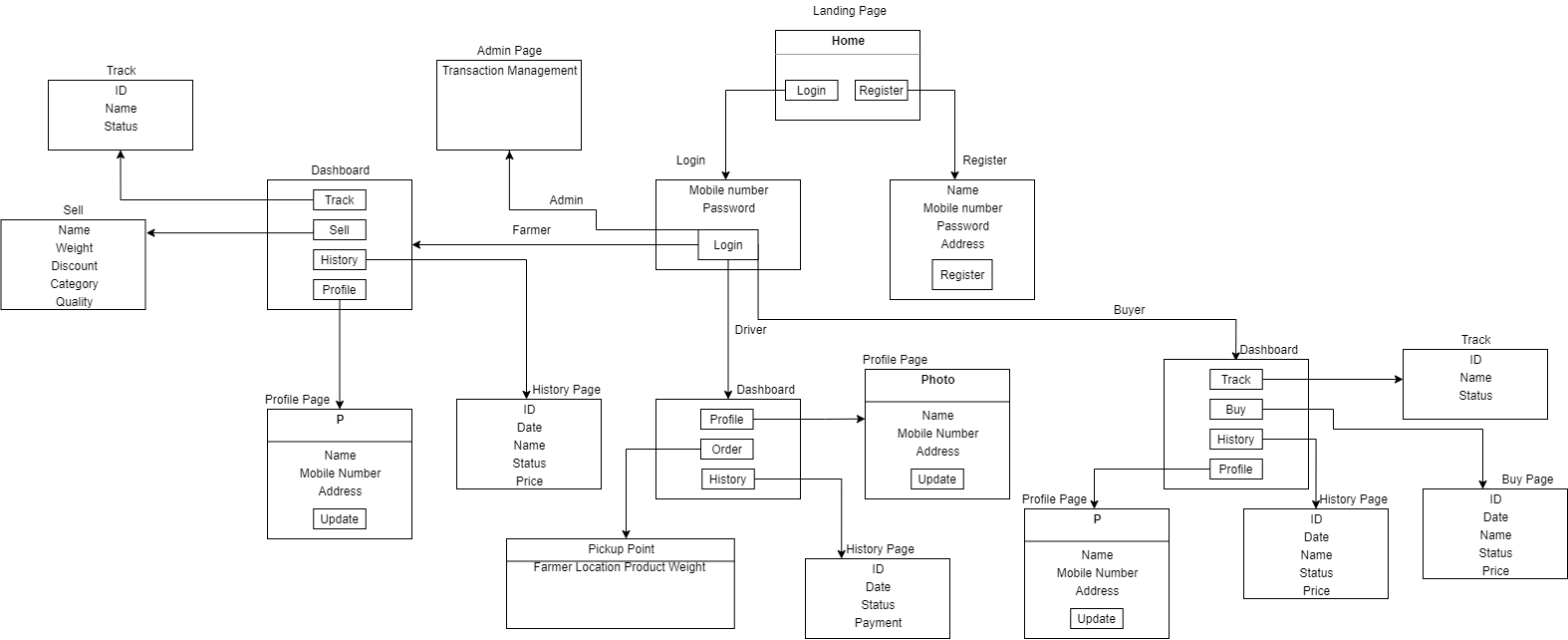
1. The first goal which we will try to achieve is to remove the unnecessary cycle in between the demand supply chain.We will try to create such a system where there will be direct buyers such as supermarts or a bulk buyer where we will directly supply the production from the farmers.Our cycle will try include maximum areas of production as we will also try to include transport services.

2. The second goal which will try to implement is to create a contract based production which will be provided by the companies. In this type what we are trying to achieve is that we will take certain contracts from the companies If the farmer decides to accept the contract then the company will provide the materials required for proper production. The company will also provide the details of the materials they will provide to the farmers for the proper production of the raw material.

* 1. **References**
* <https://flutter.dev/docs>
* <https://docs.mongodb.com/>
* <https://expresses.com/en/5x/API.html>

1. **Overall Description**
   1. **Product Perspective**

As we know that India’s GDP has a major share of the agricultural tasks and activities (around 14-16%), it becomes very important to improvise in the agricultural sector and maintain its sustainability in the fast growing market. But due to lack of advancements in this sector there is a transition seen in the farmers count in which many of them have opted to stop their farming work and move on to some other sector. Hence we have chosen this project as it is a step towards the overall agricultural sector development by helping the farmers to gain what they deserve. Kisaan Bandhu will be mainly focussed to provide a platform for the farmers to sell their products and get the maximum share of their produce.



* 1. **Product Functions**

**•User**: Farmer

**•Functions**:Farmers are one of the end user who can post the products over the application.

* Sell products
* Track history
* Track current orders
* Profile

**•User** : Buyer

•**Functions** :Buyer is one of the end user who can buy the products over the application.

* Sell products
* Track history
* Track current orders
* Profile

**•User** : Driver (Transport service)

•**Functions** :

* Track history
* Track current orders
* Profile
  1. **Operating Environment**

Operating Environment for Kisaan Bandhu is listed as below :

* Operating System : Windows/MAC/Linux and Android
* Database : MongoDB(NoSQL)
* Platform : Flutter, Nodejs, Python
* Software : postman, emulator, git, VSCode, Android Studio

Hardware Requirements :

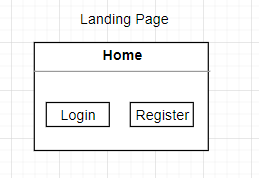
* Processor : Pentium 4 and higher
* RAM : 1GB +
* Storage : 1GB +
  1. **Design and Implementation Constraints**
* The most important factors determining whether customers return to an application are ease of use and the presence of user-friendly features. Usability testing is important for finding problems and improvements in an application. Methods for evaluating usability include heuristic evaluation, cognitive walkthrough, and user testing. Each technique has its own characteristics and emphasizes different aspects of the user experience.
* **Pages details:**
* Home/Landing Page
* Login Page
* Registration Page
* Profile Page
* Individual Dashboard Page
* Selling/Buying Product Page
* History Page
  1. **User Documentation**

Not Applicable

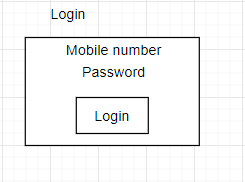
* 1. **Assumptions and Dependencies**
* **Google Maps (Internet connectivity)**
* **Working API**
* **Location Services Enabled device**

1. **External Interface Requirements**
   1. **User Interfaces**

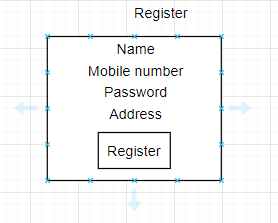
* Home/Landing Page



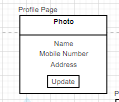
* Login Page



* Registration Page



* Profile Page

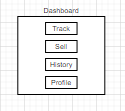


* Individual Dashboard Page

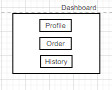
Buyer Dashboard



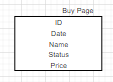
Farmer Dashboard

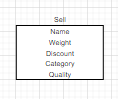


Driver Dashboard

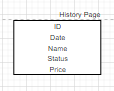


* Selling/Buying Product Page





* History Page



* 1. **Hardware Interfaces**

**Server Side:**

**The API will be hosted on a web hosting server.**

**Client side:**

1. **Android mobile**
2. **Proper Internet Connectivity**
3. **GPS (Location) enabled service**
   1. **Software Interfaces**

**Server Side:**

1. Web server will accept all the requests from the client and respond accordingly.Database will be hosted centrally using a single storage system.

**Client Side:**

1. Android OS having version 5 and above.
   1. **Communications Interfaces**

**The HTTP or HTTPS protocol(s) will be used to facilitate communication between the client and server.**

1. **System Features**

This section provides a requirement overview of the system.

* 1. **SF1**

**Title -** User Registration

**Description -** User can register in the application through mobile where the user must provide the required information.

**Dependency** - None

**Result** - User register successfully

* 1. **SF2**

**Title -** User Login

**Description -** User can Login the application through mobile where the user must provide the required information to successfully login.

**Dependency** - SF1.

**Result** - User Login successfully

* 1. **SF3**

**Title -** Sell product ( by Farmer)

**Description -** Farmers can sell the product via the application by listing their products over the application for sale.

**Dependency** - SF2

**Result** - Successfully listed the product for sale

* 1. **SF4**

**Title -** User Profile

**Description -** User can view and edit his/her profile in the application **Dependency** - SF2

**Result** - User profile changed/viewed successfully

* 1. **SF5**

**Title -** Track orders

**Description -** User can track the current orders which are placed and can get all the details regarding the status of the order.

**Dependency** - SF2 & SF3

**Result** - User will be able to track their orders successfully

* 1. **SF6**

**Title -** Track History

**Description -** Users can track their previous orders which are sold or bought and already completed and can get the status of those particular orders.

**Dependency** - SF2 & SF3

**Result** - User will be able to track their history successfully

* 1. **SF7**

**Title -** Buy Product ( Buyers)

**Description -** Users can buy the product via the application which are listed in the application for sale.

**Dependency** - SF2

**Result** - User will be able to buy products successfully

* 1. **SF8**

**Title -** track pickup orders (Drivers)

**Description -** The transport service driver will be able to track the pickup points from where the farmers will sell their products.

**Dependency** - SF2 & SF3

**Result** - Driver will able to track the pickup route successfully

1. **Other Nonfunctional Requirements**
   1. **Performance Requirements**

* This system is for a smaller area initially and hence it should handle at least 200 users.
* The downtime for the application must be less than 5%.
  1. **Safety Requirements**
* If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.
  1. **Security Requirements**
* System will use secured database
* Avoiding misuse of personal data of users such as GPS location.
* System will have different types of users and every user has access constraints.
  1. **Software Quality Attributes**
     1. **Reliability**

The system provides storage of all databases on redundant computers with automatic switchover.

The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes.

* + 1. **Maintainability**

NoSQL is used for maintaining the database and the server takes care of the API’s. In case of a failure, a re-initialization of the program is recommended as soon as possible.

* + 1. **Availability**

The system should be available at all times, meaning the user can access the application, only restricted by the down time of the server on which the system runs.

* + 1. **Portability**

The technologies used are already system independent. The application is Android based. So The end-user part is fully portable and any system using any android version 5 and above should be able to use the features of the system.