Software Requirements Specification and Report

for

Mandi Master

Prepared by

200901097 - Muhammad Mobeen (Team Leader)

200901012 - Fasih Muhammad Virk

200901124 - Ahmed Rohail Awan Waheed Yousaf

200901019 - Muhammad Hamza Farooq

200901037 - Muhammad Awais Afzal

Organization: Markhorverse (markhorverse.com)

Date: 03/15/2023

Table of Contents

1. Introduction

This report presents a Mandi Master website designed to facilitate the buying and selling of livestock and food products. The website provides a user-friendly interface for farmers, ranchers, and consumers to connect and engage in transactions. The primary goal of the website is to provide a convenient and efficient way for users to buy and sell livestock and food products including dairy products directly, thereby reducing the need for intermediaries and increasing profits for producers. The website features a real-time inventory tracking, and a messaging system for users to communicate with each other. The website also provides valuable insights into market trends and prices, allowing users to make informed decisions about their transactions. Overall, this website offers a unique solution for streamlining the livestock and food product market, benefiting both buyers and sellers alike.

1.1 Purpose

The purpose of Mandi Master is to become the medium between seller and buyer to sell and buy livestock, food, and dairy products. It allows all users to post ads regarding which item they want to sell or to buy, user can also put favorite items like which he will buy later in watch list. The overall goal of the website is to make it easier for user to sell or buy products on market price without being scammed by third parties.

1.2 Sustainable Development Goal

Our project is aligned with following sustainable goals:

- Industry innovation infrastructure.
- Economic Growth
- No poverty

1.3 Product Scope

Mandi Master website is a tool designed to help users with efficient and convenient way by selling or buying their products fast less expensive and on market rate. The website has a user-friendly interface that allows users to easily navigate and understand.

Primarily, the website will provide a feature for storing important data such asthe data related to deals that took place between buyer and seller, which can also be related to the corporate goal of keeping records and data organized and easily accessible.

In summary, the Mandi Master website is an efficient, user-friendly, and cost-saving tool that helps users to buy and sell products easily, by providingthem with a search bar, watch list, past deals history, new deals notifications. It is aligned with the corporate goals of reducing operational costs, increasing efficiency, and organizing records.

2. Design and Implementation:

2.1 Design Pattern

The Mandi Master website will use the Model-View-Controller Model (MVC) design pattern. MVC model is used for user interfaces, data and controlling logics.

The Model:
The Model represents the data and the business logic of the website. In the case of the Mandi Master website, the Model would include items such as Livestock, Crops, Fruits, Vegetables, Dairy products. Data Storage, which would contain all data and logic related to the products and previous deals.

The View:
The View represents the user interface of the website. In the case of the Mandi Master Website, theView would include the different screens and layouts that the user interacts with, such as the home screen, the dashboard, searched items screen and the previous deals data storage screen.

The Controller:
The controller acts as a bridge between the Model and the View. It is responsible for exposing the data from the Model to the View, and for handling user input and updating the Model accordingly. In the case of the Mandi Master website, the View Model would handle tasks such as displaying the watch list and storing previous deals data. It would also handle user input and update the Model accordingly.

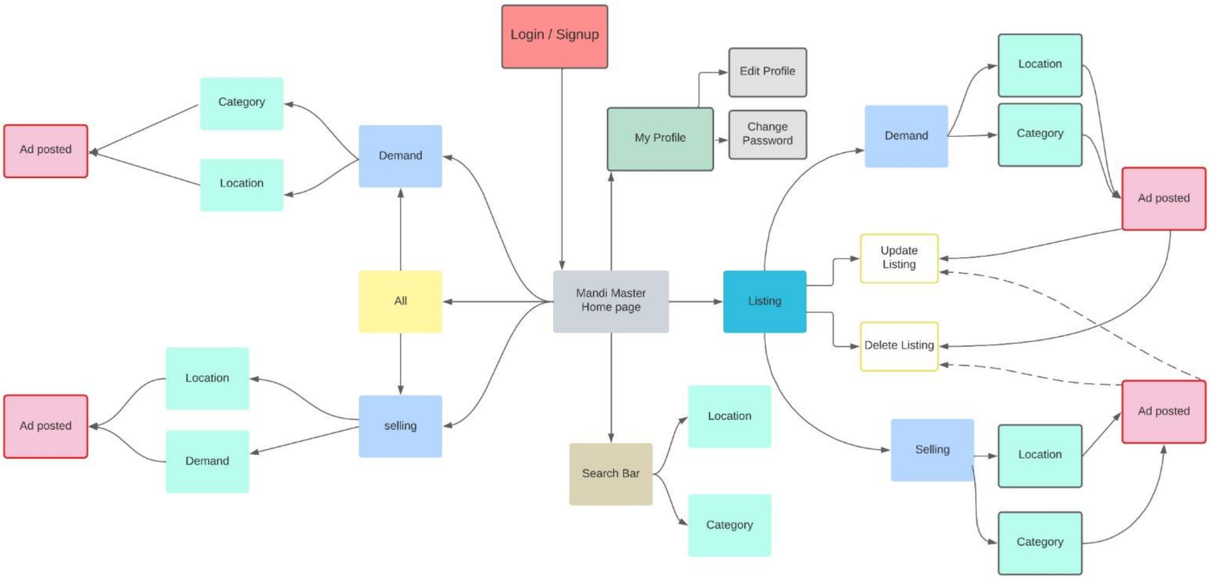
2.2 Software Architecture

The Mandi Master website is a new, self-contained product that is being developed to help users to sell and buy livestock, crops, fruits and dairy products and to keep track of their previous deals data. The website is designed to be easy to use and navigate, with a user-friendly interface that is consistent across different platforms and devices.

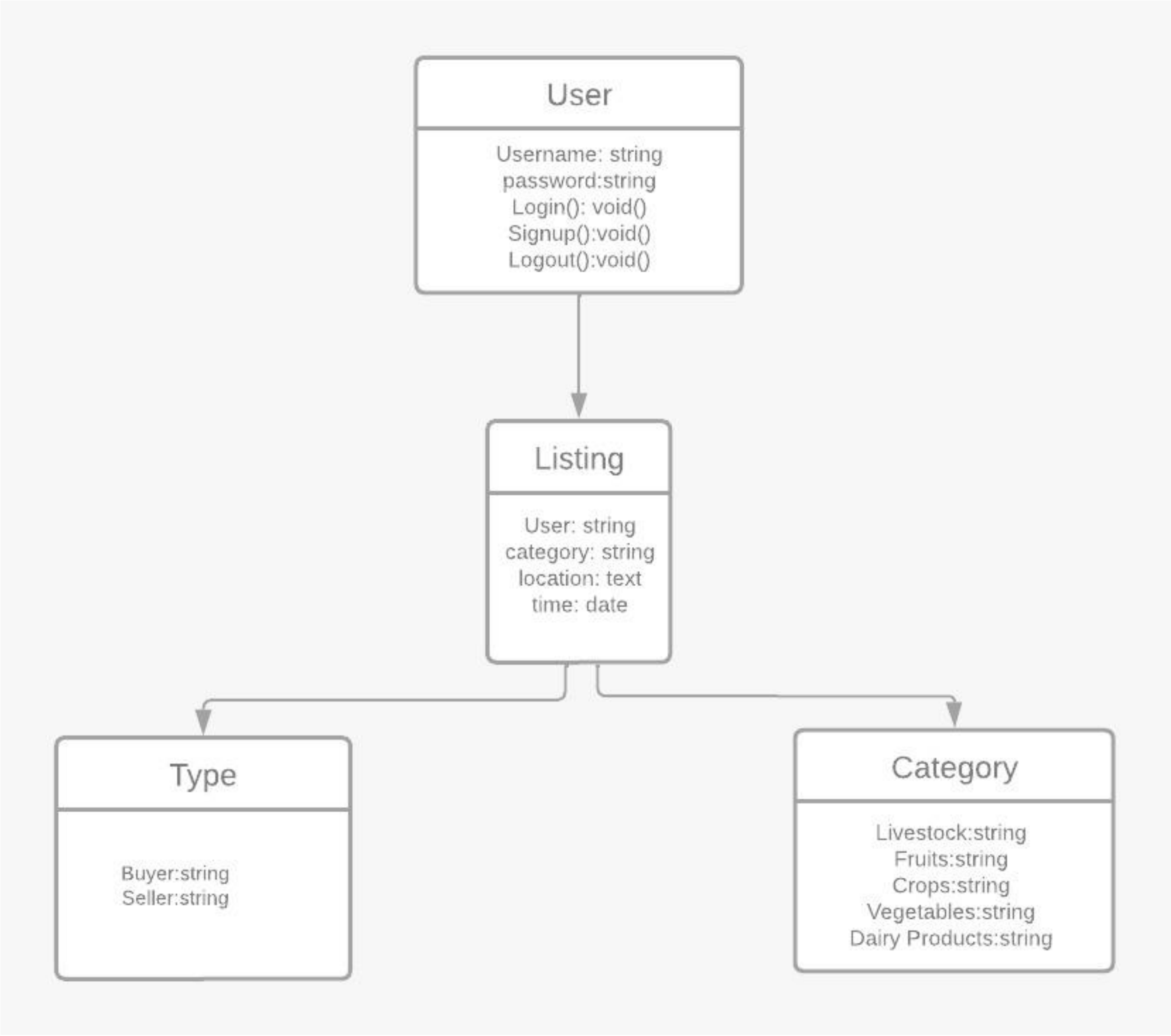
The website is the main component of the system and is directly connected to the user. The website also connects to cloud-based storage service to store and retrieve data related to the user's requirement.
In terms of origin, the product is a new, self-contained product that is being developed to fill a gap in the market for a website that help users to sell and buy livestock, crops, fruits, and dairy products and to keep track of their previous deals data.
In terms of the Class Diagram:

- **Login/Signup Class**: As our client approaches our website, first thing is to take Login/Signup action. This is mandatory due to keep record of user and for some security reasons.
- **Types Class:** There are 2 types of client modes are present in our website, you can be a buyer, or you can be a seller. All deals are rounded between these 2 keys items.
- **Categories:** In Mandi Master website we are dealing in Livestock, crops, fruits, vegetables, dairy products.
- **Listings:** In this section all the Ads will be shown.
- **Data Storage class**: This class represents the storage of user details, previous and upcoming deals details.
- **Website Settings class**: This class represents the settings of the website and contains data related to the user's preferences (e.g., enable/disable push notifications)

2.2.1 ER Diagram



2.2.2 UML Diagram



These classes are connected by the relationships of composition, association, and aggregation to show how they are related and dependent on each other. The website interfaces with the cloud-based storage services to store and retrieve data and the user interfaces with the website to manage buy and sell livestock, crops, fruits, and dairy products and to keep track of their previous deals data.

2.3 Product Functions and Use Cases

The Mandi Master website is designed for farmers, ranchers, and consumers to connect and engage with each other so to buy and sell food products within market rates and eradicate scams. There are several user classes that we anticipate will use this product:

1. Frequent Users: These are users who use the website on a regular basis to buy or sell their product. They are likely to have large stock of that product and use the website to manage all of them. These users are likely to have a high level of technical expertise and be familiar with websites.
2. Occasional Users: These are users who use the website occasionally to buy or sell their product. They are likely to have small stock of their product and use the website to manage it. These users are likely to have a moderate level of technical expertise and may be less familiar with websites.
3. Novice Users: These are users who are new to the website and may not have experience with websites. They may need assistance in understanding how to use the website and may require additional guidance and support.

The most important user class for this product are the frequent users, as they are likely to use the website on a regular basis and will be the most impacted by any issues or bugs on the website. They are also most likely to provide feedback and suggestions for improvement. The occasional users and novice users are also important, but their needs may be slightly different, and they may require additional support and guidance.
It's important to note that the website should be user-friendly and accessible to all users, including those with disabilities. The website's interface should be easy to navigate and understand and should be designed to be intuitive for users with varying levels of technical expertise.

2.4 Operating Environment

The Mandi Master website will operate on both iOS and Android devices in the web browser like Chrome.

2.4.1 Hardware Platform:

iOS: Any kind of iPhone, iPad, and iPod touch devices will be entertained.
Android: Any kind of smartphones and tablets will be entertained as we are dealing with rich and with poor peoples also.

2.4.2 Operating System and Versions:

iOS: Any kind of iOS acceptable.
Android: Any kind of Android.

2.4.3 Other software components or websites with which it must peacefully coexist:

Cloud-based storage service is used for data storage, so the website will need to integrate with these services.
The website is integrated with push notification.
The website will integrate with the device's calendar to set reminders for upcoming deals meeting with buyer or seller.
The website is designed in a way so that it will peacefully coexist with these services and websites and should not interfere with the performance of the device or other websites running on the device.

2.5 Design and Implementation Constraints

There are several issues that limit the options available to our team while creating the Mandi Master website. These include:

2.5.1 Corporate or regulatory policies:

The website must adhere to corporate or regulatory policies regarding data storage and privacy. This may limit the options available for data storage and retrieval.

2.5.2 Hardware limitations:

The website has to consider the hardware limitations of the devices it will run on, suchas memory and processing power. This may limit the options available for data storage and retrieval, as well as the features that can be included on the website.

2.5.3 Interfaces to other websites:

The website integrates with cloud-based storage services, push notification services, and the device's calendar. This limits the options available for data storage and retrieval, as well as the features that can be included on the website.

2.5.4 Specific technologies, tools, and databases to be used:

The website is developed using specific technologies, tools, and databases that are compatible with both iOS and Android devices. This limit the options available for data storage and retrieval, as well as the features that can be included in the website.

2.5.5 Parallel operations:

The website is designed to handle parallel operations to avoid delays or errors while executing multiple tasks.

2.5.6 Language requirement:

The website is developed in a programming language that is compatible with both iOS and Android devices e.g., Django framework.

2.5.7 Communications protocols:

The website uses specific communications protocols to integrate with cloud-based storage services, push notification services, and the device's calendar.

2.5.8 Security considerations:

The website is designed with robust security features to protect user data and prevent unauthorized access.

2.5.9 Design conventions or programming standards:

The website is developed following design conventions and programming standards that are compatible with both iOS and Android devices.

2.6 Assumptions and Dependencies

There are several assumed factors that could affect the requirements stated in the SRS for the Mandi Master website. These include:

2.6.1 Third-party or commercial components:

The website will be developed using third-party or commercial components such as cloud- based storage services, push notification services, and the device's calendar. The project could be affected if these components are not available or do not meet the website requirements.

2.6.2 Development or operating environment:

The website is developed for both iOS and Android devices. The website could beaffected if there are issues with the development or operating environment that prevent the website from working correctly on one or both platforms.

2.6.3 Constraints*:*

The website considers the hardware limitations of the devices it will run on, such as memory and processing power. The project could be affected if these constraintsare not met.

2.6.4 Dependencies on external factors:

The website will depend on the availability of cloud-based storage services, push notification services, and the device's calendar. The project could be affected if these services or websites are not available or do not meet the project's requirements.

It's important to note that these assumptions should be regularly reviewed and updated throughout the project's lifecycle to ensure that they are accurate and in line with the current project's status, and that any changes in these assumptions are communicated effectively to all relevant stakeholders.

3. External Interface

3.1 User Interfaces

The Mandi Master website have a user-friendly interface that is easy to navigate and understand. The interface will consist of several screens and components, including:

3.1.1 Login/Signup:

When user install the website, first he has to sign up and then login in to the website.

3.1.2 Home screen:

The home screen will display the search bar, number of ads related to previous search, my profile, and dashboard. It will also display button of buying and selling where user is buying or selling its products.

3.1.3 Dashboard:

Dashboard will display two buttons. One is buying and other one is selling along with search bar.

3.1.4 History:

This will display user’s previous deals data.

3.1.5 My profile:

This will display user his/her profile where he/she manages his/her profile.

3.1.6 Settings screen:

The settings screen will allow users to customize the website's settings, such as enabling or disabling push notifications, and managing their account information.

The website follows the guidelines and standards of the respective platform (iOS or Android) for GUI and product family style and includes standard buttons and functions such as a "Help" button and a "Settings" button on every screen. Error message display standards is also in line with the respective platform's guidelines.

The website includes keyboard shortcuts for easy navigation. For example, users will be able to use keyboard shortcuts to access different sections of the website, such as selling items list or buying items list.

The user interface design is documented in a separate user interface specification, which will include detailed information on the layout, design, and functionality of each screen and component.

3.2 Hardware Interfaces

The Mandi Master website will interface with the hardware components of the device on which it is installed. The logical and physical characteristics of these interfaces include:

3.2.1 Supported device types:

The website is compatible with both iOS and Android devices, including smartphonesand tablets.

3.2.2 Nature of data and control interactions:

The website interacts with the device’s storage and memory to store and retrieve data, such as the maintenance schedule, service history, and fuel consumption data. The website interacts with the device's calendar and push notification services to set reminders and send notifications to users.

3.2.3 Communication protocols:

The website uses standard communication protocols, such as HTTP/HTTPS, to communicate with cloud-based storage serves and push notification services. The website usesStandard APIs provided by the device’s operating system to interact with the device's calendar and storage.

3.2.4 Input/Output:

The website accept input from the user through the device's touch screen and keyboard and will display output through the device's screen.

3.2.5 Security:

The website uses standard encryption protocols such as SSL/TLS to secure the communication between the website and the cloud-based storage services and push notification services.

Overall, the website will interface with the device’s hardware components in a logical and efficient manner, using standard communication protocols and APIs to ensure smooth and secure data interactions, and providing a seamless user experience.

3.3 Software Interfaces

The Mandi Master website will connect to several other specific software components,including:

3.3.1 Cloud-based storage services:

The website uses a cloud-based storage service, firebase to store and retrieve data such as the user’s data, deals history, watch list items. The website uses standard HTTP/HTTPS protocols to communicate with the storage service.

3.3.2 Push notification services:

The website use push notification services for iOS and Android to send push notifications to users. The websitewill use standard APIs provided by these services to send and receive notifications.

3.3.3 Device's calendar:

The website will use the device's calendar to set reminders for upcoming deals either to buy or sell products. The website will use standard APIs provided by the device’s operating system to interact with the calendar.

3.3.4 Data items or messages coming into the system:

The website will store and retrieve data from the cloud-based storage service and the user interfaces with the website to manage buy and sell livestock, crops, fruits, and dairy products and to keep track of their previous deals data. The website will also receive push notifications from the push notification service.

3.3.5 Data items or messages going out of the system:

The website will send data the cloud-based storage service. The website will also send push notifications to users through the push notification service.

3.3.6 Services provided:

The website provides services such as data storage and retrieval, push notifications, and calendar integration.

3.3.7 Nature of communications:

The website uses standard HTTP/HTTPS protocols to communicate with the cloud-based storage service. The website will use standard APIs provided by the push notification service andthe device's operating system to send and receive notifications and interact with the calendar.

3.3.8 Data sharing mechanism:

The website will use standard HTTP/HTTPS protocols to share data with the cloud-based storage service, and standard APIs provided by the push notification service and the device'soperating system to share data with the push notification service and the calendar.

The detailed website programming interface protocols for these software components will be described in separate documents. Any data sharing mechanism must be implemented in a way that ensures data integrity, security, and privacy of the users.

3.4 Communications Interfaces

The Mandi Master website will require several communications functions, including:

3.4.1 Cloud-based storage service communications:

The website uses standard HTTP/HTTPS protocols to communicate with the cloud-based storage service for data storage and retrieval. The website will use XML data formatting fordata transfer.

3.4.2 Push notification service communications:

The website will use standard APIs provided by the push notification service to send and receive notifications. The website will use the service's standard data formatting for data transfer.

3.4.3 Network server communications:

The website will use standard HTTP/HTTPS protocols to communicate with the network server fordata storage and retrieval.

3.4.4 Communication standards:

The website will use standard HTTP/HTTPS protocols for communication with the cloud-based storage service, push notification service, and network server.

3.4.5 Communication security:

The website will use standard encryption protocols such as SSL/TLS to secure the communication between the website and the cloud-based storage service and push notification service.

3.4.6 Data transfer rates:

The website will transfer data at a rate that is website for the device's hardware and network connection.

3.4.7 Synchronization mechanisms:

The website will use standard synchronization mechanisms to ensure that data is up to date and consistent across all devices.

Overall, the website's communication functions will be designed to provide secure and efficient data transfer, with a focus on ensuring data integrity, security, and privacy of the users. The detailed communication protocols, encryption and security mechanisms will be described in separate documents.

System Requirements

3.5 Requirement F1

Create an authentic platform for buyers and sellers.

3.5.1 Description and Priority

This website should allow users to make their deals with ease, task that are included in our website are like buying and selling of crop, life stock, grocery items, animal feed etc. Users should be able to set reminders for upcoming deals.

3.5.2 Input/Outputs Sequences

- Input: User selects “Buy” or “Sell” button according to its need.
- Output: User is prompted to enter his asset information.
- Input: User enters information about his product.
- Output: Deal will be done on the spot or dealing schedule is created and displayed on the home screen.

3.6 Requirement F2

Track on going deals between buyer and seller.

3.6.1 Description and Priority

The website should allow users to track their deals. If stamp paper is allocated for deal, then this website will remind the buyer to full fill the deal before stamp paper got expired. This can cause heavy penalty on the buyer.
Priority: High

3.7 Requirement F3

Store important documents related to the deal. If a big deal of crop is done, then this website will maintain the data of this deal on his data base.

4.1.1 Description and Priority

The website should allow users to store important documents related to their specific deal, such as buying and selling of life stock, crop, or grocery item.
Priority: Medium

4.1.2 Input/Outputs Sequences

- Input: User selects "Store deal" button
- Output: User is prompted to select the deal to be stored
- Input: User selects the specific deal to be stored
- Output: Deal is stored and displayed on the document storage screen

3.8 Requirement F4

Customize website settings.

4.1.1 Description and Priority

The website should allow users to customize the website's settings, such as enabling or disabling push notifications, and managing their account information.
Priority: Low

4.1.2 Input/Outputs Sequences

- Input: User selects "Settings" button
- Output: User is prompted with the settings options
- Input: User selects the setting they want to change
- Output: Selected setting is changed and saved

4. User Stories and Scenarios

4.1 Functional Requirement: Create a stable platform for buyers and sellers.

- User Story: As an website holder, I want to sell camel for 100k then as I post the Ad this must to shown in Dashboard so that I can keep track of my deal.

- ID: PK-001
- Scenario:
  1. The user opens the website then home screen will be loaded.
  2. The user taps on the "Sell" button, a prompt will appear then chose “camel”.
  3. The website prompts the user to enter their animal information (e.g., age, breed, color).
  4. The user enters the animal information and taps "Next".
  5. The user adds tasks to the schedule and sets reminders for upcomingdeal if someone message to meet him.
  6. The website saves the dealing schedule and displays it on the home screen.

4.2 Functional Requirement: Store important documents related to the deal.

- User Story: As a website owner, I want to be able to store important documents related past deals (e.g., transaction amount, date, person’s name) so that I can have easy access to them.
- ID: PK-003, PK-100
- Scenario:
  1. The user opens the website and is taken to the home screen.
  2. The user taps on the "Store documents" button. This will store all the document related to the deal.
  3. The website prompts the user to select the document to be stored.
  4. The user selects the document and taps "Save".
  5. The website stores the document and displays it on the document storage screen.

4.3 Functional Requirement: Customize website settings.

- User Story: As a user, I want to be able to customize the website's settings (e.g., enable/disable push notifications) so that I can have a personalized experience.
- ID: PK-004
- Scenario:
  1. The user opens the website and is taken to the home screen.
  2. The user taps on the "Settings" button.
  3. The website displays the settings options (e.g., push notifications, account information).
  4. The user selects the setting they want to change.
  5. The website updates the selected setting and saves the changes.
  6. The user goes back to the home screen and verifies that the setting has beenchanged successfully.

6. Nonfunctional Requirements

6.1 Performance:

The website must be able to perform all its functionalities smoothly and efficiently, even when handling large amounts of data. The acceptable response time for all functionalities must be less than 2 seconds.

6.2 Scalability:

The website must be able to handle an increasing number of users without any significant degradation in performance.

6.3 Security:

The website must protect the user's data and personal information from unauthorized access. All data transfer between the website and cloud-based storage services and network server, must be encrypted and all sensitive data must be stored in encrypted form.

6.4 Usability:

The website must be easy to use and navigate, with a user-friendly interface. The user interface must be consistent across different platforms and devices. The website must be accessible to all users, including those with disabilities.

6.5 Compatibility:

The website must be compatible with different devices and operating systems (iOS and Android) and must run on the latest versions of these operating systems.

6.6 Maintenance:

The website must be easy to maintain and upgrade. The website's code should be well-organized and structured and should be designed to be easily modified or extended.

6.7 Backup and Recovery:

The website must have a robust backup and recovery system in place to ensure that data is not lostin case of any system failure or unexpected event.

6.8 Internationalization:

The website must be able to support multiple languages and must be able to display different languages based on the user's preference.

6.9 Compliance:

The website must comply with all laws and regulations, including data privacy laws.

6.10 Reporting:

The website must be able to generate different types of reports related to the buying and selling contracts between 2 parties, about what they bought or what they sold. These reports should be exportable in different formats (e.g., PDF, CSV, Excel)

6.11 User-Friendly:

The website must be -friendly and should work smoothly on different screen sizes and resolutions.

7. Test Plan

The testing strategy for the Mandi Master website will include a testing method to ensure that the website meets the requirements and functions as intended. The testing process will involve multiple stages, including unit testing, integrationtesting, system testing, and acceptance testing.

7.1 Unit testing:

Unit testing will be used to test individual components of the website, such as the listing of the product on the dashboard, buying any product, selling any product, proper Ads of their product. This will be done by creating test cases for each component and verifying that they function as expected.

7.2 Integration testing:

Integration testing will be used to test the interactions between different components of the website, such as the integration between the buyer and seller contract. This will be done by creating test cases for different scenarios and verifying that the website behaves as expected.

7.3 System testing:

System testing will be used to test the website, including its performance, usability, and compatibility with different devices and operating systems. This will be done by creating test cases for different scenarios and verifying that the website behaves as expected.

7.4 Acceptance testing:

Acceptance testing will be used to test the website with real users, to ensure that it meetstheir needs and expectations. This will be done by creating test cases for different scenarios and verifying that the website behaves as expected.

Overall, the testing strategy for the Mandi Master website will focus on verifying that the website functions as intended and meets the requirements specified in the functional requirements. The Test Traceability matrix will be used to map each test case to one or more requirements, and the testing coverage results will be used to measure the effectiveness of the testing process.

8. Futures Plans:

We will implement all these features soon.

8.1 Watchlist:

When a user visit Mandi Master app, and he likes to purchase anything later he can add this item into watchlist whenever he likes to make deal, he will access it with ease.

8.2 Dealer section:

When a successful deal happens, and they want to work in future. They chat with each other and run their business accordingly.

8.3 Ads and paid posting:

Once our client base becomes strong, we will start to include Ads and Paid Posts, this will help us to earn revenue.

8.4 Always space for scalability:

In the earn future, our client ratio on our server will increase tremendously, so at that moment we can easily scale our website for best performance.