Mandi

Develop an application for Apple sellers to sell their Produce in the nearby Mandi (marketplace). The price of their Produce depends on factors like the village price per unit, weight, and loyalty index.

- There is a set of Villages that has a selling price set per kg of the Produce in INR. Each village is uniquely identified by a name. The price is set to two decimal precision points.
    - Example: **Ramnagar** has a selling price of **120.08 INR per kg**
- There is a set of registered Sellers identified uniquely by their name. Each of them have an associated unique Loyalty card ID.
    - Example: Seller ***Ramu Kaka*** has a loyalty card ID ***S18972***
- There can be registered and unregistered Sellers. Unregistered Sellers do not have Loyalty card ID.
- Autofill the loyalty card info, based on the registered seller name.
- Autofill the registered seller name if the loyalty card number is entered.
- The loyalty Index for registered Sellers is 1.12 and for unregistered Sellers is 0.98. The index gets multiplied by the calculated price.
- Calculate the price of the sellable produce based on the data available to you from test data and input by the App user.

Expectations:

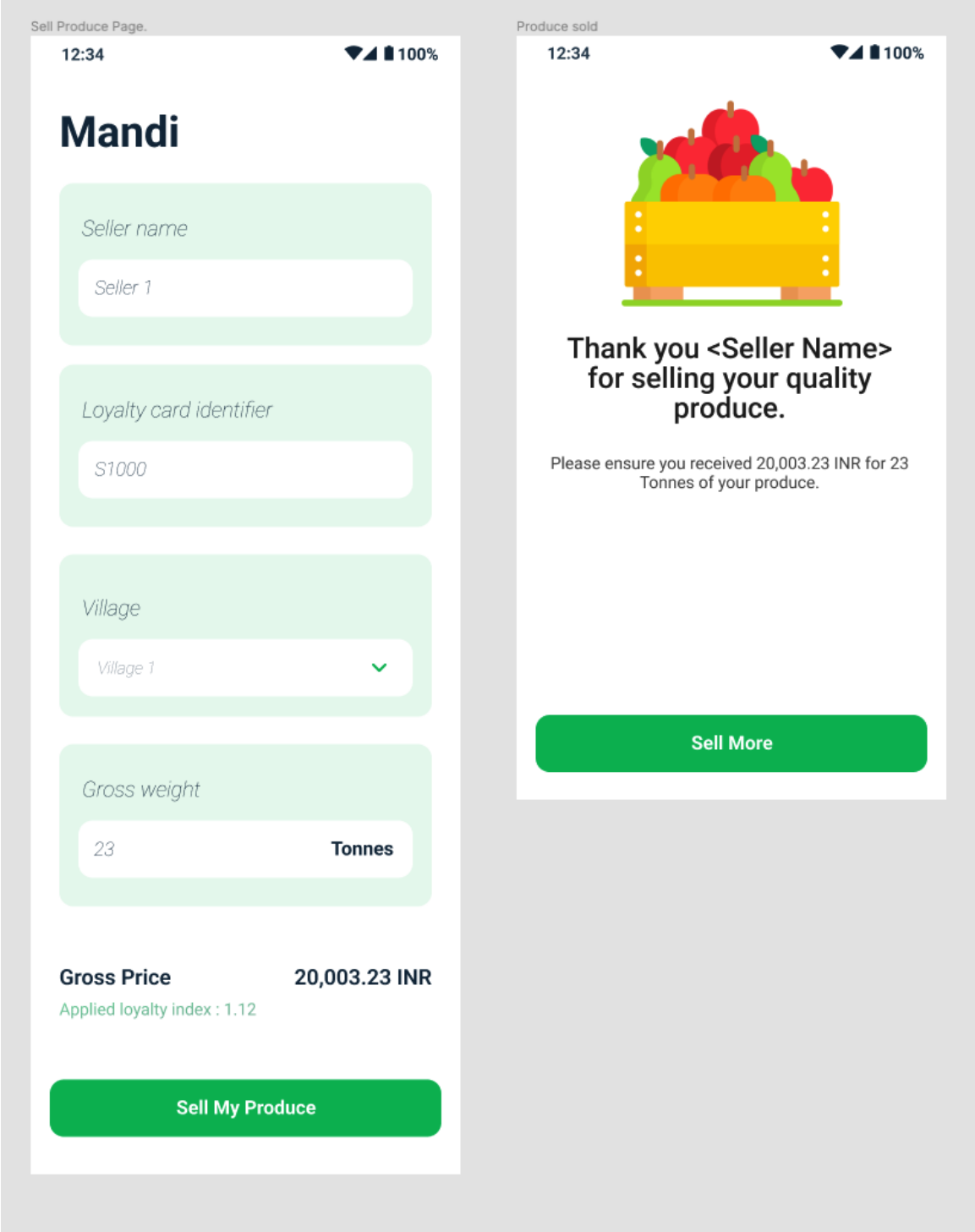
1. Clean, readable, and modular code with good separation of concerns: Like us humans, our code also needs to be sociable. Your fellow developers should like interacting with it.
2. Use of the latest Android architecture libraries, conventions, and guidelines is encouraged.

3. Easy to run: Anyone who checks out your code for the first time, should have all the information necessary to run it.
4. Idiomatic code: If you are using Kotlin, we need to understand if you really know Kotlin and decided to use the language because of the things it offers over Java. Please don't write Kotlin like Java.
5. Clear responsibilities: Heard about the Single Responsibility Principle?
6. Have the necessary models to better facilitate information flow between classes.
7. Unit tests / Integration tests: Don't like mocking? no worries, we're all in for a Mockists vs Classicist debate 😇. Regardless, we value having comprehensive tests.
8. Intelligent and efficient use of libraries: Be ready to explain why you thought you needed to use a particular library 🙂
9. Intuitive directory structure: Any new person who looks at your code should be able to easily navigate their way through the packages/directories.
10. For anything not mentioned in the problem statement, make appropriate assumptions and mention them in the **README** for documentation purposes.

Things to note:

- DO NOT copy other people's code
- DO NOT make this problem statement or your solution publicly available
- The mockups provided for the assignment are just for guidance, please use your creativity to design the app the way you see fit.

Wireframes



> fruit box icon: https://www.flaticon.com/premium-icon/fruits_2194917