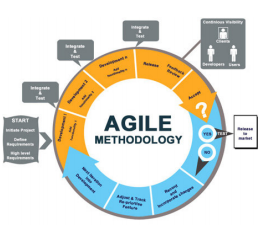
**CHAPTER 4: IMPLEMENTATION**

**4.1Model:**

Agile: A suitable software process model for Mobile Application development. Agile software development uses iterative development as a base but it differs from other traditional software process models as it has more customer-centric viewpoint. Agile process model uses continuous customer feedback to successively refine and deliver a software system.



**Figure 4.1.1: Agile Methodology**

It follows a combination of iterative and incremental approach where the entire SDLC is broken into small iterations which help the project to adapt to changes rapidly. It minimizes overall risk. It gives importance to Customers and their feedback. It believes in self-organizing teams and customer collaboration for developing an amply satisfied product. Numerous cycles of testing and quality assurance increases reliability.

The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

The most popular agile methods include Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). These are now collectively referred to as Agile Methodologies, after the Agile Manifesto was published in 2001.

Following are the Agile Manifesto principles −

* Individuals and interactions − In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
* Working software − Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
* Customer collaboration − As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
* Responding to change − Agile Development is focused on quick responses to change and continuous development.

Agile methods are being widely accepted in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the Agile model.

The advantages of the Agile Model are as follows −

* Is a very realistic approach to software development.
* Promotes teamwork and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* Suitable for fixed or changing requirements
* Delivers early partial working solutions.
* Good model for environments that change steadily.
* Minimal rules, documentation easily employed.
* Enables concurrent development and delivery within an overall planned context.
* Little or no planning required.
* Easy to manage.
* Gives flexibility to developers.

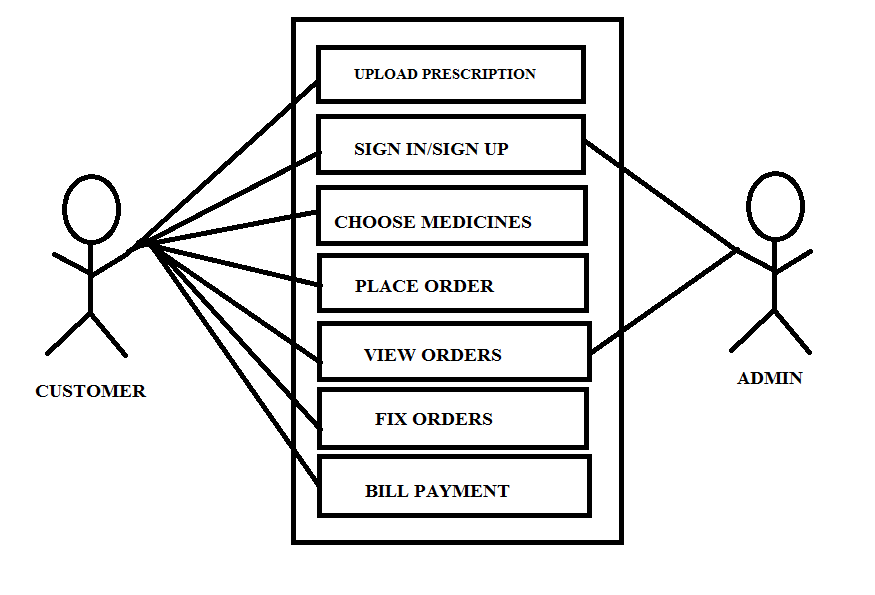
The disadvantages of the Agile Model are as follows −

* Not suitable for handling complex dependencies.
* More risk of sustainability, maintainability and extensibility.
* An overall plan, an agile leader and agile PM practice is a must without which it will not work.
* Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
* Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
* There is a very high individual dependency, since there is minimum documentation generated.
* Transfer of technology to new team members may be quite challenging due to lack of documentation.

When to use Agile model:

* When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
* To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
* Unlike the [waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/) in agile model very limited [planning](http://istqbexamcertification.com/what-is-the-purpose-and-importance-of-test-plans/) is required to get started with the project. Agile assumes that the end users’ needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly affected or removed based on feedback. This effectively gives the customer the finished system they want or need.
* Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

**4.2: Use Case Diagram:**

****

**Figure 4.2.1: Use Case Diagram**

Description:

Customer:

1. Upload Prescription: In this module the customer uploads the prescription in which prescribed medicines are given.

2. Sign in/Sign up: In this module the customer registers himself and signs in.

3. Choose Medicines: In this module customer chooses medicines from various types of medicines.

4. Place Orders: In this module customer places his order.

5. View Orders: Customers can view orders placed.

6. Fix Orders: It is a module in which the orders are fixed.

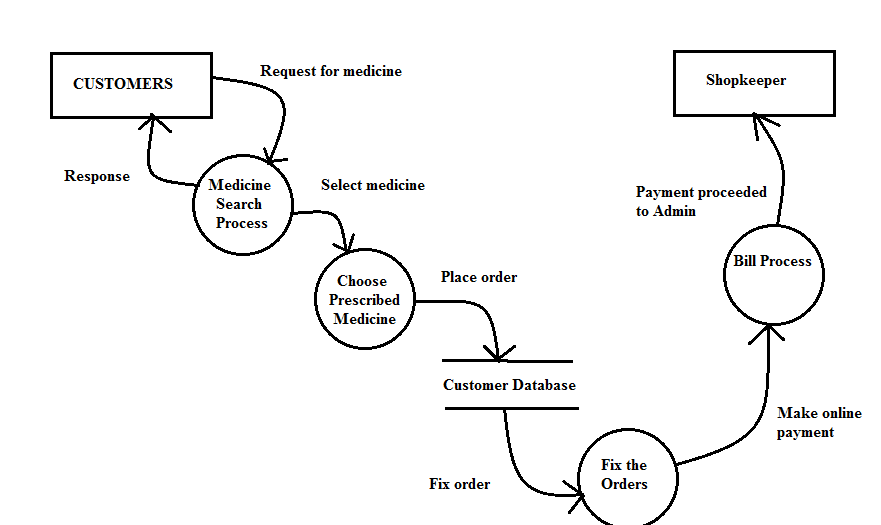
7. Bill Payment: It is a module in which customer can make online payment for fixed orders.

Admin:

1. Sign in / Signup: In this module Admin can fetch data of the authenticated user.

2. View Orders: In this module Admin can look up for the orders placed by customers.

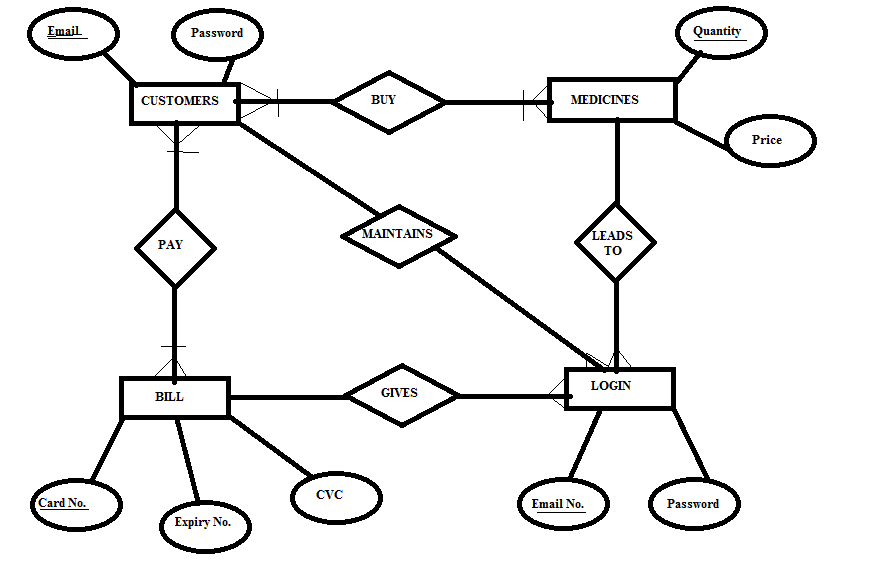
**4.3DataFlow Diagram:**

****

**Figure 4.3.1: Data Flow Diagram**

Here in this dataflow diagram the flow is very simple. The customer search for the medicine. When the customer selects his prescribed medicines from various types of medicines. The customer places the order and fixes the order. All this information is stored in customers database .After the order is fixed a bill is generated. This bill is then directed to the Admin.

**4.4 E-R Diagram:**



**Figure 4.4.1: E-R Diagram**

Description:

There are four entities in E-R diagram and five relationships. The entities are as follows:

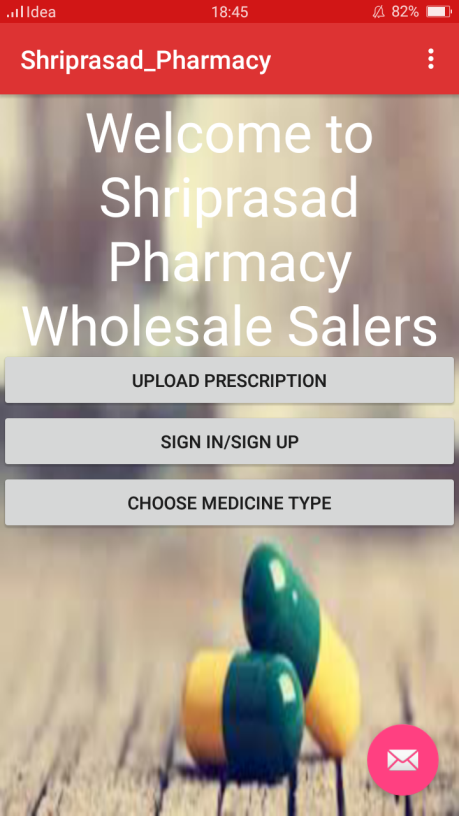
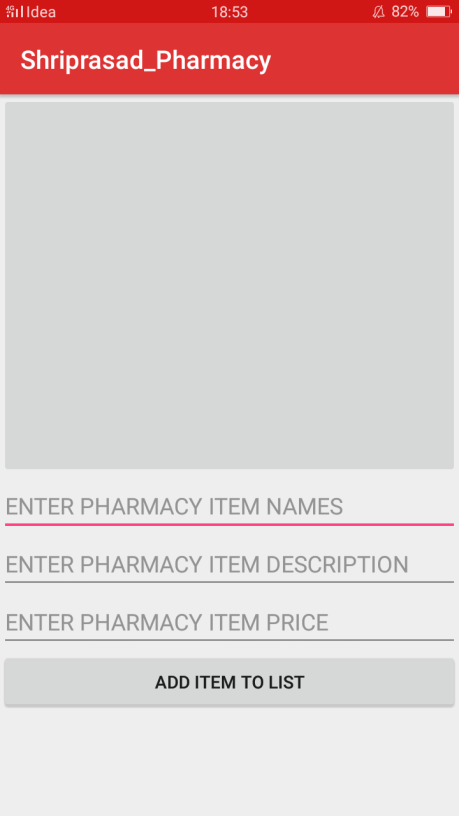
* Customers
* Medicines
* Bill
* Login

The relationships between these entities are as follows:

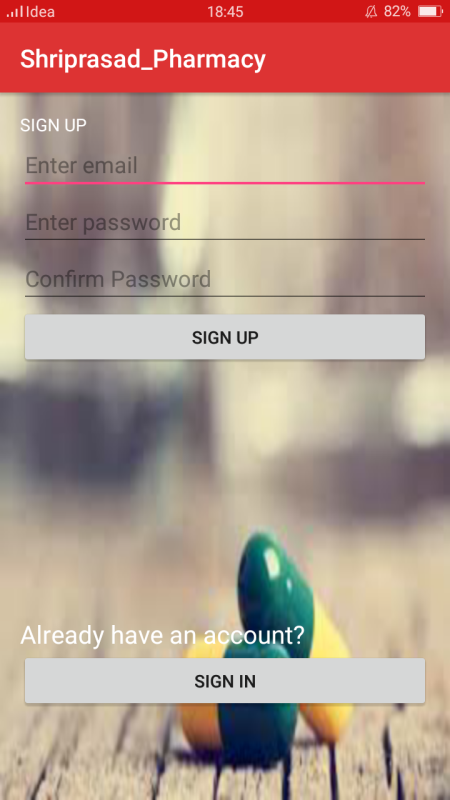
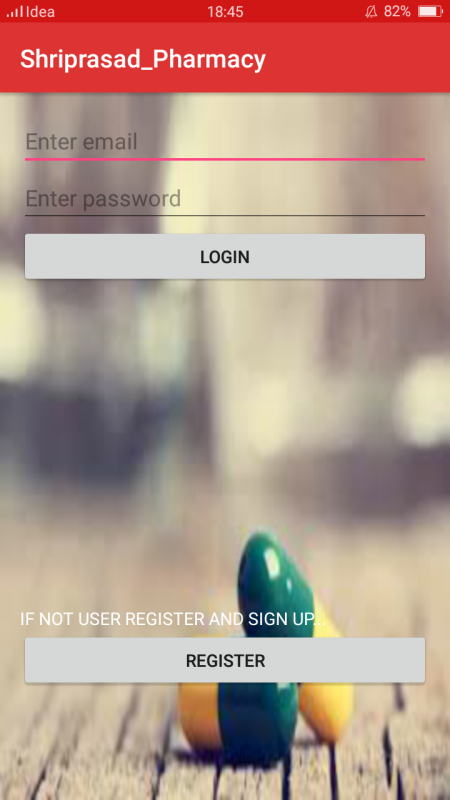
* Pay
* Maintains
* Maintains
* Gives
* Buy

Here in this E-R Diagram the customer can buy medicines when the customer is logged in. All the entities are related to the login module. The customer uploads prescription and then chooses the medicines they want. After they choose the medicine, they place and fix the medicines choose. After that for online payment the bill page is made available which has fields like card no. , expiry no., cvc.

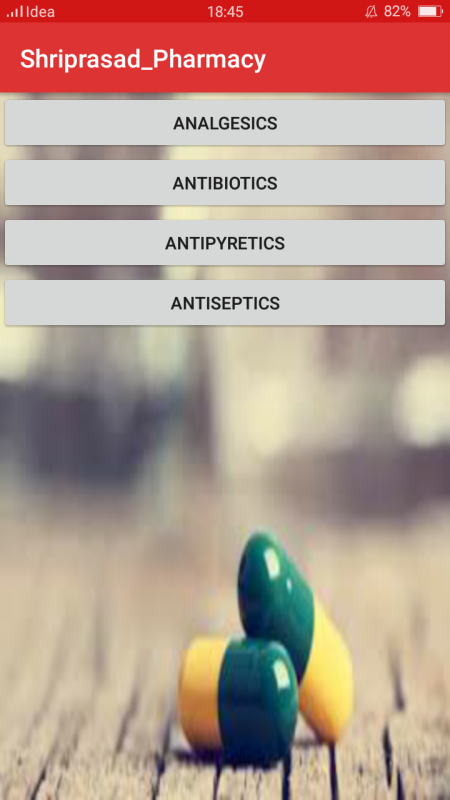
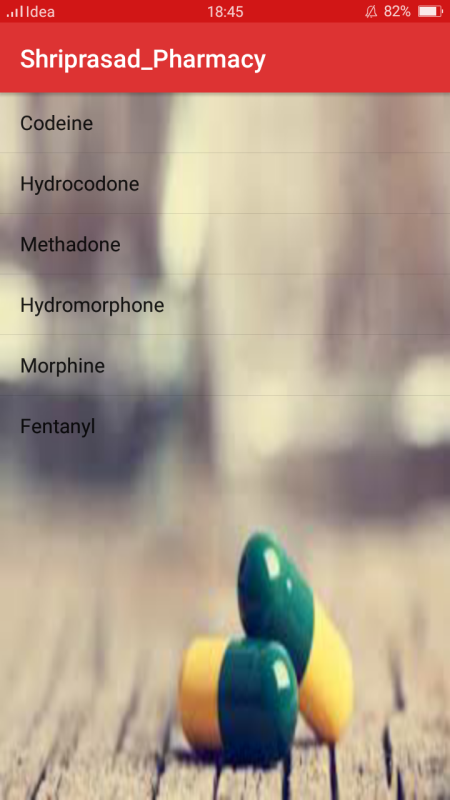
4.5 User Interface Design:

**Screen 4.5.1: Launcher page Screen 4.5.2: Upload Prescription page**

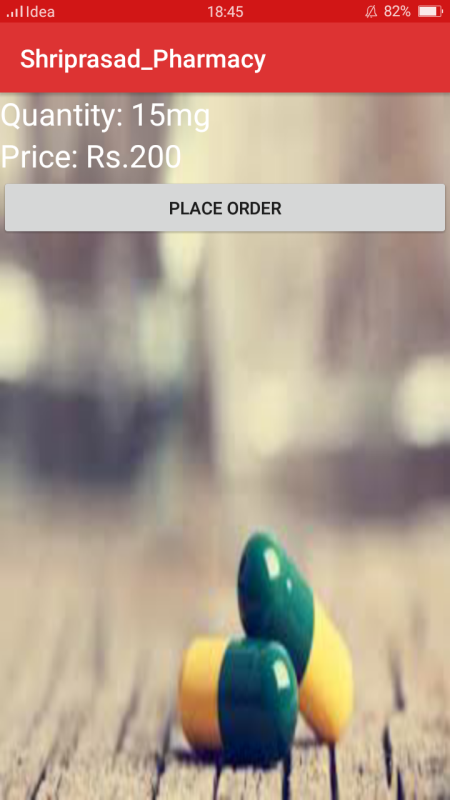
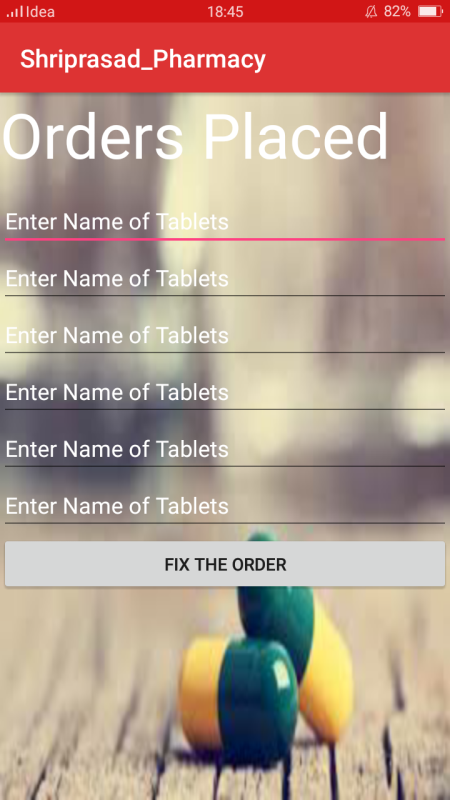
** **

**Screen 4.5.3: Signup page Screen 4.5.4: Login page**

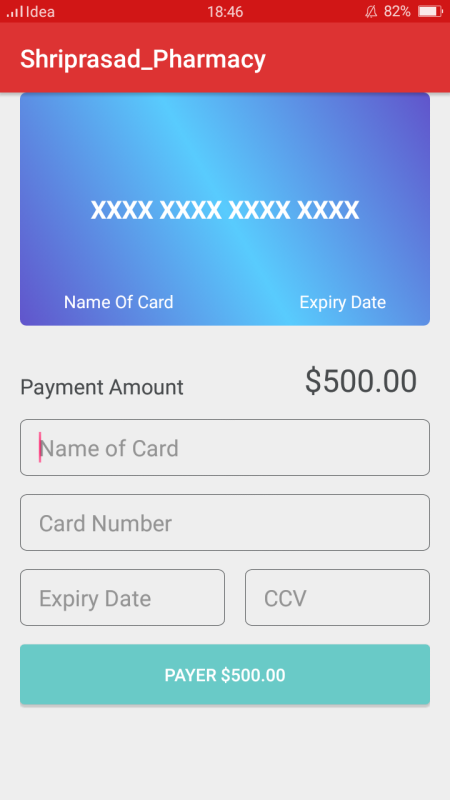
** **

**Screen 4.5.5: Choose Medicine Screen 4.5.6: List of Particular**

**Type page Type of medicines**

** **

**Screen 4.5.7: Information of Tablet Screen 4.5.8: Fix the orders**

****

**Screen 4.5.9: Billing page**