

# Software Product Line Engineering Project

CS 415

## Phase 2 Report

**Group Name: Seniors**



### Team Members

<b>Names</b>	<b>IDs</b>
Berat Biçer	21503050
Ezgi Çakır	21402334
Gülce Karaçal	21402016
Metehan Kaya	21401258
Seda Gülkesen	21302403

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Software Product Line Engineering course CS415.

# Table of Contents

<b>1. Introduction.....</b>	<b>3</b>
<b>2. Requirements .....</b>	<b>3</b>
<b>3. Kano Methodology.....</b>	<b>3</b>
<b>4. Variants.....</b>	<b>10</b>
4.1 FeedMe .....	10
4.2 MaturePear.....	10
4.3 HealthyLife .....	11
<b>5. Features.....</b>	<b>11</b>
5.1 Common Features .....	11
5.2 Variable Features .....	12
5.3 Customized Features.....	13
<b>6. Feature Model.....</b>	<b>14</b>
<b>7.Variability Configuration.....</b>	<b>14</b>
7.1 FeedMe .....	14
7.2 MaturePear.....	14
7.3 HealthyLife .....	15
<b>8. High Level Architecture .....</b>	<b>15</b>
8.1 Use Case Model .....	15
8.1.1 Payment Method Use Case Model .....	15
8.1.2 Pricing Use Case Model .....	16
8.1.3 Collect Points Case Model.....	16
8.1.4 Customer Support .....	17
8.1.5 Set Service Threshold .....	18
8.2 Component Diagram.....	19
8.3 Class Diagram .....	20
<b>9. Prototype .....</b>	<b>21</b>
9.1 Common Prototypes .....	21
9.2 Variable prototypes.....	23
9.3 Software Reuse in Prototype .....	27
<b>10. Discussion.....</b>	<b>28</b>
10.1 Requirements .....	28
10.2 Feature Model .....	29
10.3 Kano Methodology.....	29
10.4 Orthogonal Model .....	30
10.5 Prototype.....	30
<b>11. Source Code.....</b>	<b>31</b>
<b>12. Group Member Contribution .....</b>	<b>31</b>

# 1. Introduction

OrderLine is the order management system that is implemented for the mobile e-commerce companies so that their target user can get service through the application. The order management system is planned to be used as a basis for 3 online service ordering applications that are: Food Order & Delivery application named FeedMe, Service Order application named MaturePear and Online Pharmacy application named HealthyLife. In this report, the proposed variants will be explained. Results of kano methodology, feature model of those variant, use cases diagrams, and class diagram can be found in the rest of the report. Moreover, the most functional pages of those variants can be found in the last part of the report.

## 2. Requirements

The main motivation of the software product line engineering is to provide customized products at reasonable costs [1]. Since the order management system has considerable amount of reusable parts, it is chosen for this project, and software product line engineering strategies have been applied throughout the semester. The common functionalities are ordering a single product, filtering, searching, having minimum service limit, giving feedback, displaying chat history, providing customer support with limited working hour, checking availability of service and online payment.

## 3. Kano Methodology

Kano Model is developed in order to satisfy customer needs and yield high customer satisfaction. Questionnaire is prepared and tested on 5 members of the group to get different responses which provide different perspective. By applying results of the kano model that involves different answers, high-quality products can be served. The functionalities and dysfunctionalities are determined by 5 choices which are “Like”, “Expected”, “Do not care”, “Can live with it”, “Dislike”. According to the results, classifications are made. The results of Kano Models are below.

Meal Order Application						
Questions	Like	Expected	Do not care	Can live with it	Dislike	Classification
User collects points when he orders meal	4	1				Satisfier

User does not collect point when he orders meal			1	1	3	
User can get discount that requires some points	5					Satisfier
User cannot get discount that requires some points			1	1	3	
App provides lottery that requires some points to join	4		1			Delighter
App does not provide lottery that requires some points to join			4	1		
User loses his points when they are not used for a month					5	Undesired
User does not lose points when they are not used for a month	5					
Listed services can be displayed according to user's location	5					Satisfier
Listed services cannot be displayed according to user's location				1	4	
User can add a person as a friend	3	1	1			Satisfier
User cannot add a person as a friend			2		3	
App provides leaderboard shows points of your friends	3	1	1			Satisfier
App does not provide leaderboard shows points of your friends	1		1		3	
User can share leaderboard on social media	1		4			Indifferent
User cannot share leaderboard on social media			4		1	
App can provide weekly leaderboard	1	1	3			Indifferent
App cannot provide weekly leaderboard			4		1	
App can provide monthly leaderboard	1	1	3			Indifferent
App cannot provide monthly leaderboard			3		2	
App can provide annual leaderboard	1	1	3			Indifferent
App cannot provide annual leaderboard			3		2	
User can be informed about the stock availability	4	1				Satisfier
User cannot be informed about the stock availability			1	1	3	
User can see his friends' activities	3	1	1			Delighter
User cannot see his friends' activities				3	2	
User can demand addition of new ingredients to products	1	1	3			Indifferent
User cannot demand addition of new ingredient to products				3	2	
User can decide arrival date of the service	4	1				Satisfier
User cannot decide arrival date of the service					5	
User can check stock availability	4	1				Satisfier
User cannot check stock availability					5	

**Figure 1:** First questionnaire and classification results

Meal Order Application
------------------------

Questions	Like	Expected	Do not care	Can live with it	Dislike	Classification
User can pay with cash	2	3				Basic
User cannot pay with cash				1	4	
User can pay with credit card	2	3				Basic
User cannot pay with credit card				1	4	
User can pay online with his credit card	2	3				Basic
User cannot pay online with his credit card				1	4	
User can pay by using PayPal	1		4			Indifferent
User cannot pay by using PayPal			3	1	1	
App can send verification message to pay for security issues	4		1			Delighter
App cannot send verification message to pay for security issues				4	1	
Customer can use app free	5					Satisfier
Customer cannot use app free				1	4	
Restaurant can use app free	4		1			Delighter
Restaurant cannot use app free				4	1	
Customer can rate the service	3	2				Satisfier
Customer cannot rate the service				2	3	
Customer can write review about the service	2	3				Basic
Customer cannot write review about the service				1	4	
Customer can get history of service he got	3	2				Satisfier
Customer cannot get history of service he got				2	3	
Service provider can get history of service he provided	3	2				Satisfier
Service provider cannot get history of service he provided				1	4	
App limits services with minimum cost according to location				3	2	Undesired
App does not limit services with min cost according to location	4	1				
User can get customer support	2	3				Basic
User cannot get customer support					5	
App can remove service provider with false, invalid, etc. info	2	3				Basic
App cannot remove service provider with false, invalid, etc.info				1	4	
User can report service provider with false, invalid, etc. info	2	3				Basic
User cannot report service provider with false, invalid, etc. info				1	4	

**Figure 2:** Second questionnaire and classification results

Service Order Application
---------------------------

Questions	Like	Expected	Do not care	Can live with it	Dislike	Classification
Listed services can be displayed according to user's location	3	2				Delighter
Listed services cannot be displayed according to user's location				3	2	
User can chat directly with the service provider	2	3				Basic
User cannot chat directly with the service provider				2	3	
User can negotiate price of the service via online chat	3	1	1			Delighter
User cannot negotiate price of the service via online chat				4	1	
User can advise service provider via social media	4		1			Delighter
User cannot advise service provider via social media			2	3		
User can get unlimited number of services	5					Satisfier
User cannot get unlimited number of services				1	4	
User can sue service provider by using app	4	1				Delighter
User cannot sue service provider by using app				4	1	
App can provide lawyer to customers	4	1				Delighter
App cannot provide lawyer to customers				4	1	
App can provide lawyer to service providers	4	1				Delighter
App cannot provide lawyer to service providers				3	2	

**Figure 3: Third questionnaire and classification results**

Service Order Application						
Questions	Like	Expected	Do not care	Can live with it	Dislike	Classification
User can pay with cash	1	4				Basic
User cannot pay with cash				1	4	
User can pay with credit card	2	3				Basic
User cannot pay with credit card				2	3	
User can pay online with his credit card	3	2				Delighter
User cannot pay online with his credit card				3	2	
User can pay by using PayPal	3	1	1			Delighter
User cannot pay by using PayPal			1	3	1	
App can send verification message to pay for security issues	5					Delighter
App cannot send verification message to pay for security issues			1	3	1	
Customer can use app free	5					Satisfier
Customer cannot use app free				1	4	
Service provider can use app free	5					Delighter

Service provider cannot use app free				4	1	
Customer can rate the service	3	2				Satisfier
Customer cannot rate the service					5	
Customer can write review about the service	3	2				Satisfier
Customer cannot write review about the service					5	
Customer can get history of service he got	1	4				Basic
Customer cannot get history of service he got				1	4	
Service provider can get history of service he provided	4	1				Delighter
Service provider cannot get history of service he provided			4		1	
App limits services with minimum cost according to location			1	3	1	Undesired
App does not limit services with min cost according to location	4	1				
User can get customer support	4	1				Delighter
User cannot get customer support				4	1	
App can remove service provider with false, invalid, etc. info	2	3				Basic
App cannot remove service provider with false, invalid, etc.info				1	4	
User can report service provider with false, invalid, etc. info	2	3				Basic
User cannot report service provider with false, invalid, etc. info				1	4	

**Figure 4:** Fourth questionnaire and classification results

Online Pharmacy Application						
Questions	Like	Expected	Do not care	Can live with it	Dislike	Classification
Listed services can be displayed according to user's location		1	4			Indifferent
Listed services cannot be displayed according to user's location			4	1		
User can decide arrival date of the service	3	1	1			Delighter
User cannot decide arrival date of the service				3	2	
Any user can get notification when his order is on stock	4		1			Delighter
A user may not get notification when his order is on stock				4	1	
Only premium users can get notification about stock				4	1	Indifferent
None of users get notification about stock				3	2	
User can check stock availability	4	1				Satisfier
User cannot check stock availability					5	
App can provide email notifications	4	1				Delighter
App cannot provide email notifications				4	1	

App can provide notification via message	4	1				Delighter
App cannot provide notification via message				4	1	
App can provide medicine for animals	4		1			Delighter
App cannot provide medicine for animals			1	3	1	
Any user can get medicine for animals	4	1				Delighter
A user may not get medicine for animals			1	3	1	
App can provide locations of close hospitals	5					Delighter
App cannot provide locations of close hospitals			1	3	1	
App can provide contact info of hospitals for emergency	5					Delighter
App cannot provide contact info of hospitals for emergency			1	3	1	

**Figure 5:** Fifth questionnaire and classification results

Online Pharmacy Application						
Questions	Like	Expected	Do not care	Can live with it	Dislike	Classification
User can pay with cash	4		1			Delighter
User cannot pay with cash			4	1		
User can pay with credit card		4	1			Basic
User cannot pay with credit card			1	1	3	
User can pay online with his credit card	1	4				Basic
User cannot pay online with his credit card				2	3	
User can pay by using PayPal	4		1			Delighter
User cannot pay by using PayPal			1	3	1	
App can send verification message to pay for security issues	4	1				Delighter
App cannot send verification message to pay for security issues				4	1	
Customer can use app free	5					Delighter
Customer cannot use app free				3	2	
Customer can use app based on a subscription type		4	1			Basic
Customer cannot use app based on a subscription type			1		4	
Customer can rate the service	1	4				Basic



Customer cannot rate the service				1	4	
Customer can write review about the service	1	4				Basic
Customer cannot write review about the service				1	4	
Customer can get history of service he got		5				Basic
Customer cannot get history of service he got				1	4	
Service provider can get history of service he provided	2	3				Indifferent
Service provider cannot get history of service he provided				3	2	
App limits services with minimum cost according to location			1	1	3	Undesired
App does not limit services with min cost according to location	2	3				
User can get customer support	3	2				Satisfier
User cannot get customer support				1	4	
App can remove service provider with false, invalid, etc. info	2	3				Basic
App cannot remove service provider with false, invalid, etc.info				1	4	
User can report service provider with false, invalid, etc. info	2	3				Basic
User cannot report service provider with false, invalid, etc. info				1	4	

**Figure 6 : Sixth questionnaire and classification results**

## **4. Variants**

### **4.1 FeedMe**

FeedMe has two different types users: customer and restaurant owner. Customer login will be used by customers who want to order food. Simply, users registered as a customer can search restaurants menus based on their locations using provided filters and can order single or multiple food. The system provides payment at the door and online payment option to their users. If user chooses "pay at the door" option, then they are able to pay with credit card or cash. In addition to that, if users choose online payment, then they are able to pay with their credit card. Moreover, users can collect points based on their orders so that they can use these points to get discount for upcoming orders. Another unique function is having online social platform in which users can interact with other users, see others' actions and take a part in leaderboard alongside with their friends. On the other hand, restaurants can use restaurant login and can upload their restaurants information such as its location, its menu, its campaigns etc.

### **4.2 MaturePear**

MaturePear has two different users: customer and service provider login. First of all, with customer login, users can search for a specific type of services such as cleaning, renovation, transportation, custom furniture making, graphic design, logo design and website making etc. Also, users are able to communicate with service provider since providers will share their contact information on the website. Users can use online chatting to discuss the details of the expected service with service providers so that customers can negotiate on the price. The chat history is saved into database so that users are able to display their chat history. However, with MaturePear, users cannot make a payment at door. Just online payments with PayPal system is accepted. Furthermore, through using service provider login, providers can share their information by creating a customized website for their own work. On this website, they can create portfolio which shows their previous works, reviews for works, references and CVs. MaturePear will show a list of providers to users and by clicking on the name of the provider, users can reach the provider's customized websites.

## 4.3 HealthyLife

HealthyLife has two different users: customer and pharmacy. First of all, with customer login, users can search for a specific medicine and can order it by using an online payment with their credit cards. This application does not provide payment at door option. Also, users are able to check availability of products on different pharmacies. In addition to this feature, users will be provided a price comparison screen based on the prices of their medicines on different pharmacies. Furthermore, there are two types of customer subscribers: free and premium. With free account, users are able to perform actions mentioned above. With premium account, the system will provide additional features to users such as sending e-mail notification to users when their medicine is on stock and sending warning messages via email address before users' medicines are over. Moreover, users with premium account will be notified about several campaigns based on their previous orders. On the other hand, pharmacies can use pharmacy login and can share stock information and they can announce special deals for premium users.

## 5. Features

### 5.1 Common Features

- **Order Single Product/ Service:** All of the applications has basic functionality that user can order single product or service through the application.
- **Filter Product/Service:** All of the applications has filtering functionality that user can filter the products and services based on her/his need. It facilitates user life that s/he can easily find more suitable or affordable option.
- **Search Product/ Service:** All the applications has search functionality that user can search the products and services based on her/his need.
- **Minimum Service Limit:** All of the applications has a minimum service limitation that is determined for each service. However, the limit is designed according to work amount that service provider allocated or cost of the product or service.
- **Give Feedback:** Customers can rate the service that they got and they can write review about it.
- **Display Order History :** All user types (customers and service providers) can display the previous services that they got or they provided.

- **Customer Support in Working Hour:** All user types can get customer support using online chat bots or call. The chats are not saved so users cannot display chat history.
- **Availability of Service:** All of the applications provide users a functionality that they can check the availability of each of the product or service through application.
- **Online Payment:** All of the applications provide online payment opportunity to their users. Online Payment varies as paying with credit card and PayPal.

## 5.2 Variable Features

- **Order Multiple Products/ Services:** Users can order multiple products or services at the same time. This functionality is available for FeedMe and HealthyLife. However, in MaturePear, users are not able to get same type of service at the same time.
- **Online Chatting:** Users can chat directly with the service providers in MaturePear. They can use online chatting to discuss the details of the expected service and to negotiate the price.
- **Display Chat History:** This feature arose as a need from Online Chatting feature. All the discussions are saved into the system so users can access that chats and they can display their chat history later. Since only MaturePear has Online Chatting feature, only users who use MaturePear are able to display chat history.
- **Location based Service Provider Listing :** When the users search for the specific service, the system displays the possible service providers according to the customers location first. In FeedMe, customers can get service from the restaurants that are close to the given address. However, In HealthyLife and MaturePear, listed service providers are not location based.
- **Set Service Threshold Based on Working Hours:** MaturePear has minimum service limit based on working hours. Each service provider who uses this app can determine his/her minimum service limit. To give an example, cleaning lady can say that she will provide service minimum half of a day.
- **Set Service Threshold Based on Cost:** Both in HealthyLife and FeedMe, service providers can set service threshold. In HealthyLife, pharmacies can set threshold based on product cost. In FeedMe, each restaurant can set threshold based on distance between restaurant and customer's address.
- **Social Feed :** In FeedMe, users can add their friends, and they stay updated about their friend's activities. In MaturePear and HealthyLife, there is no social feed functionality.

- **Payment at Door :** In FeedMe users are able choose payment at door option and they can pay with their credit card or cash. HealthyLife and MaturePear does not provide paying at the door option to their users.
- **Online Payment with PayPal:** Only in MaturePear, users are able to pay with PayPal.
- **Online Payment with Credit Card:** FeedMe and HealthyLife provides online payment with credit card option to users. MaturePear does not provide paying online with credit card option to their users.
- **Pricing :** FeedMe and MaturePear is free to use by their customers. Pricing for customer usage in HealthyLife is based on subscription type: regular or premium. Premium users pay 1\$ per month to use the app. However, all of the applications charges service providers with extra fees.
- **Collect Points:** In FeedMe, users collect points and they can see their rank on the generated leaderboard among their local friend list or global users. Both in FeedMe and in HealthyLife, by collecting points users can get discount for upcoming orders.
- **Collect points to get discount :** In FeedMe and HealthyLife users can collect points when they order the meal. They can get discount for the upcoming orders.
- **Having Leaderboard:** In FeedMe, the system provides a leaderboard that compares your points with your friends and globally.

### 5.3 Customized Features

- **Premium Membership:** Healthy Life has two different subscribers: regular and premium. HealthyLife-Premium provides day&night customer support and users are notified about campaigns and special deals.
- **Day&Night Customer Support:** HealthyLife-Premium provides 7/24 online live support to their premium subscribers.
- **Get Notification:** In HealthyLife-Premium, users can get notification when their order appears in stock. In addition to that, they are notified about campaigns, and special deals based on their previous orders.

## 6. Feature Model

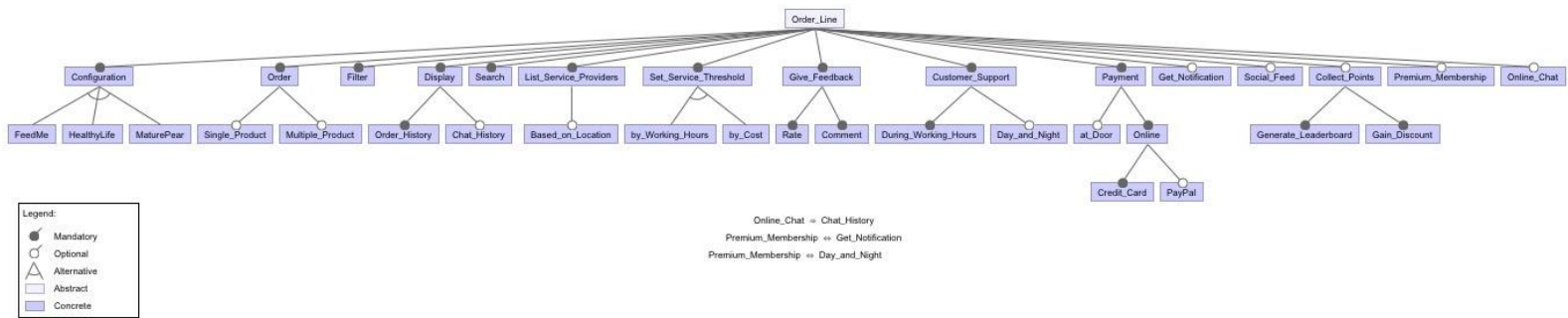


Figure 8: Feature Model of OrderLine

## 7. Variability Configuration

### 7.1 FeedMe

For FeedMe, users can benefit from following features that are marked with red star sign in feature model.

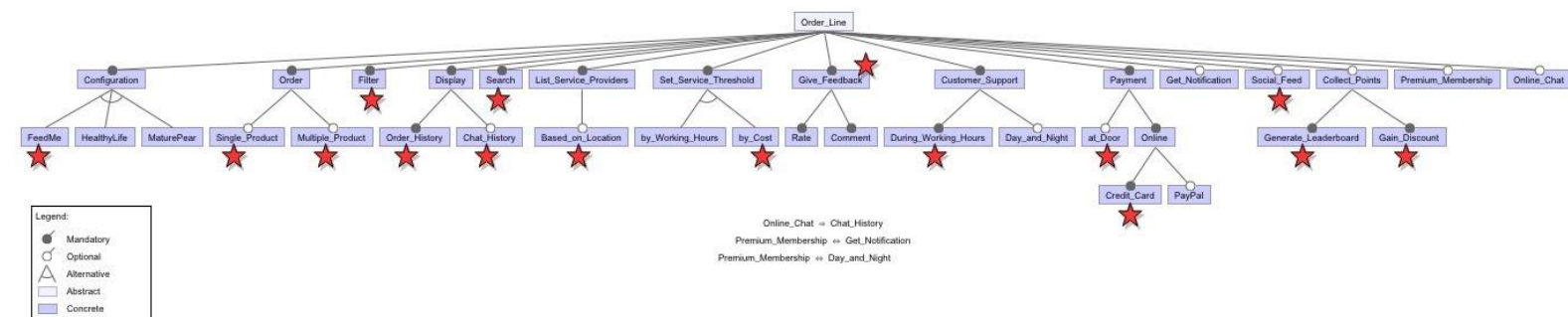
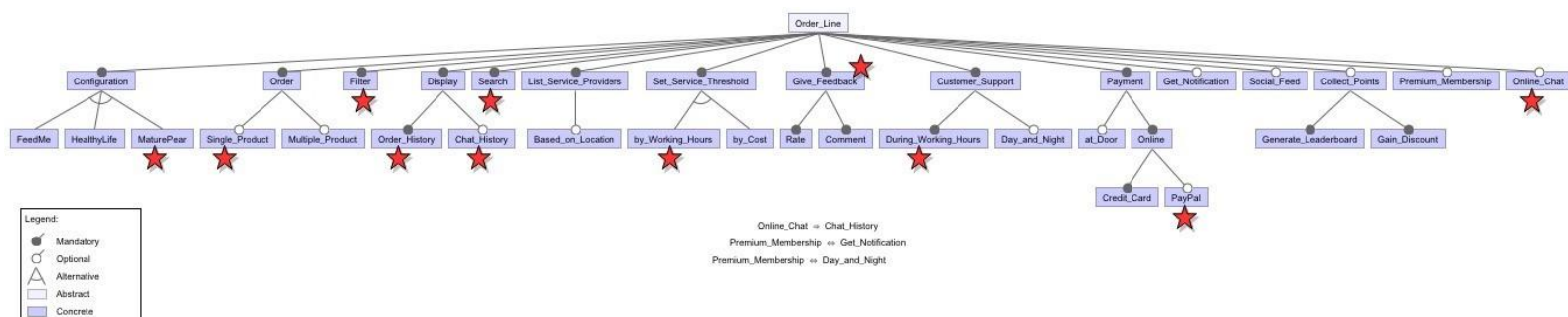


Figure 9: Feature Model of FeedMe

### 7.2 MaturePear

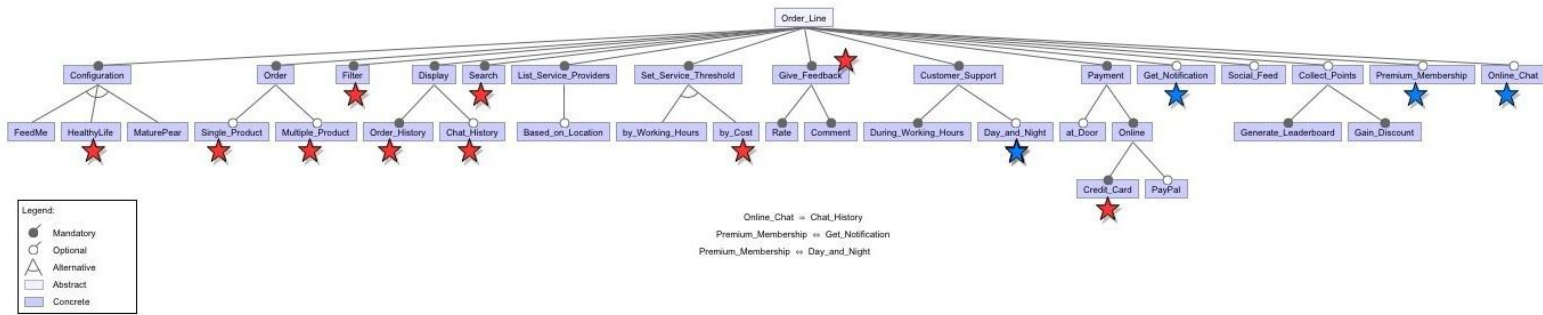
For MaturePear, users can benefit from following features that are marked with red star sign in feature model.



**Figure 10: Feature Model of MaturePear**

## 7.3 HealthyLife

For HealthyLife, users can benefit from following features that are marked with red star sign in feature model. The features that are indicated with blue star sign represents the feature of HealthyLife-Premium.



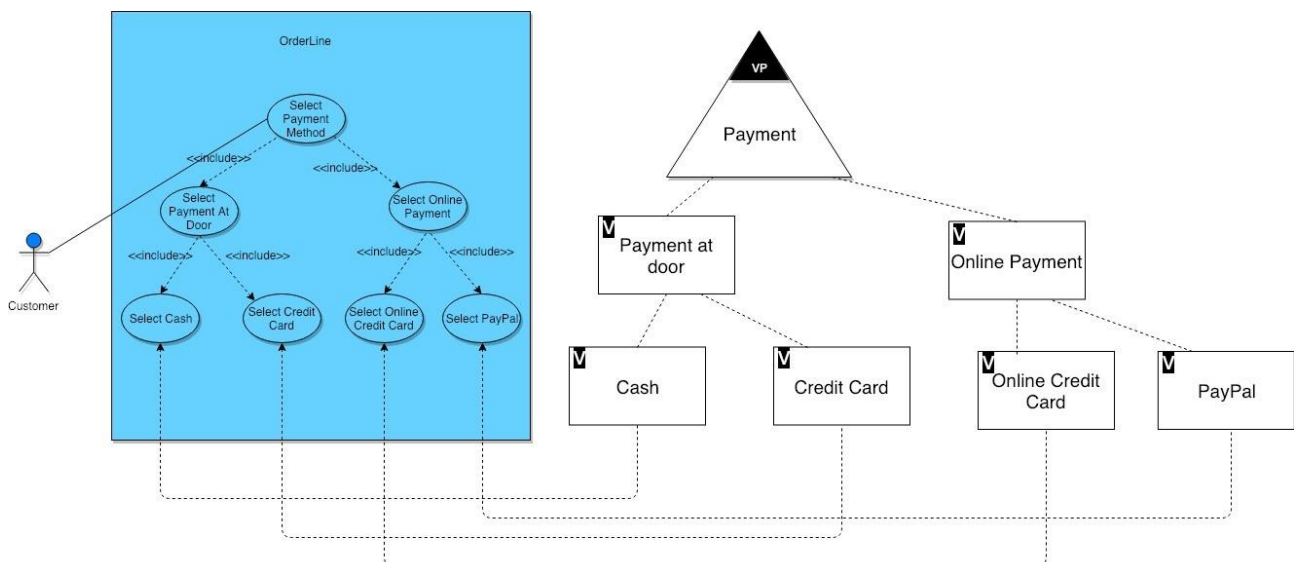
**Figure 11: Feature Model of HealthyLife**

## 8. High Level Architecture

### 8.1 Use Case Model

#### 8.1.1 Payment Method Use Case Model

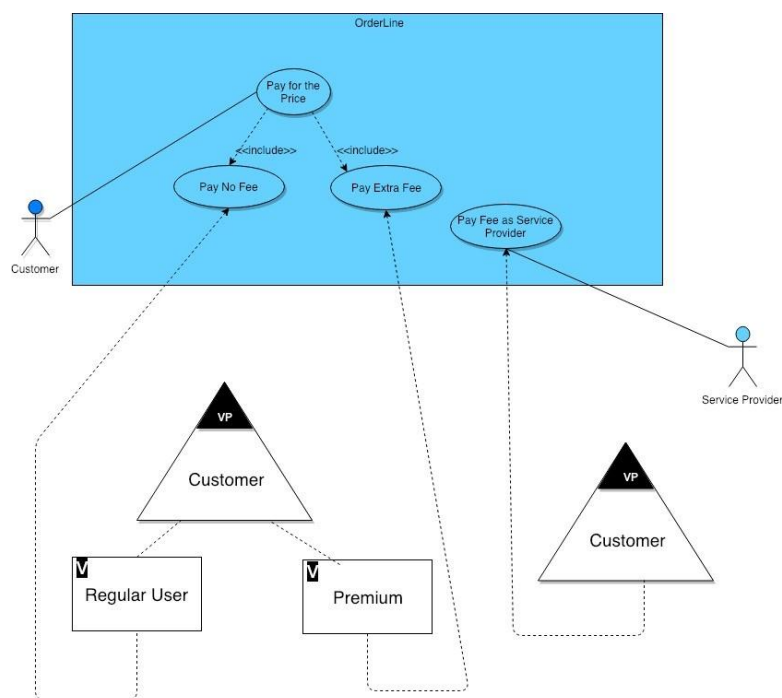
A customer can make his payment in two main different ways: at the door and online. Payment at the door includes cash and credit card. Online credit card and PayPal are the ways to pay online. FeedMe and HealthyLife provides online payment with credit card option to users. MaturePear does not provide paying online with credit card option to their users. Only MaturePear users can use PayPal method.



**Figure 12 :** Use case model for Payment Method

### 8.1.2 Pricing Use Case Model

There are different pricing options for different users like customer and service provider. FeedMe customer usage is free. However, FeedMe charges the restaurants with extra fees. MaturePear customer usage is free. However, MaturePear charges the service providers with extra fees. Pricing for customer usage in HealthyLife is based on a subscription type. Premium customers are supposed to pay 1\$ per month.

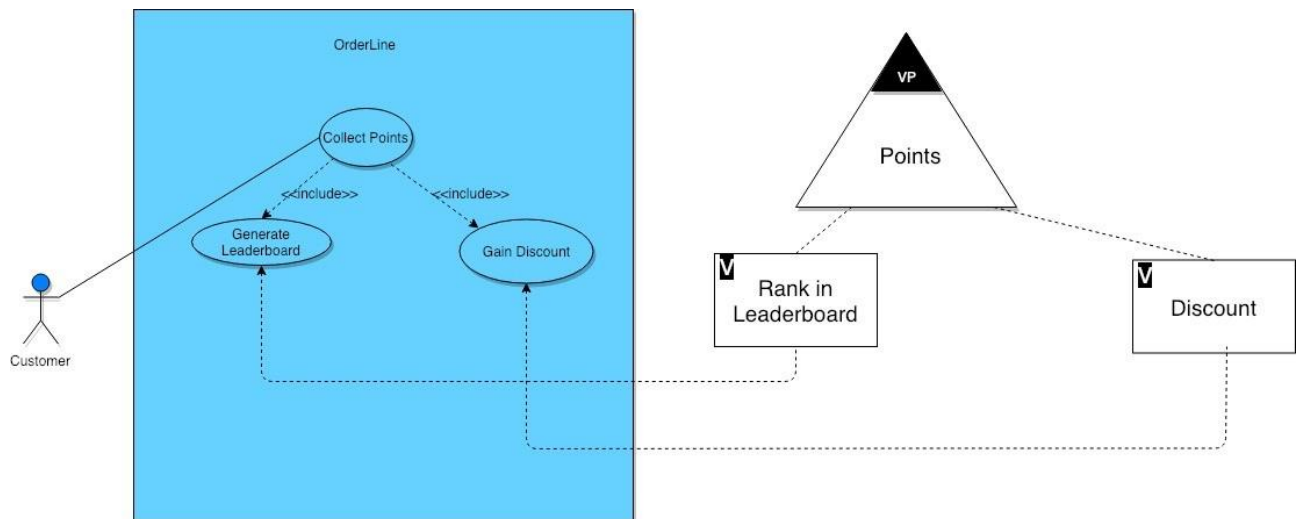


**Figure 13 :** Use case model for Pricing

### 8.1.3 Collect Points Case Model

Users may collect points as they order products in application. After they collect, they may use this points for gaining discount for their upcoming orders. Otherwise, they may be listed in a leadership list based on their points in the application. These features are available in FeedMe.

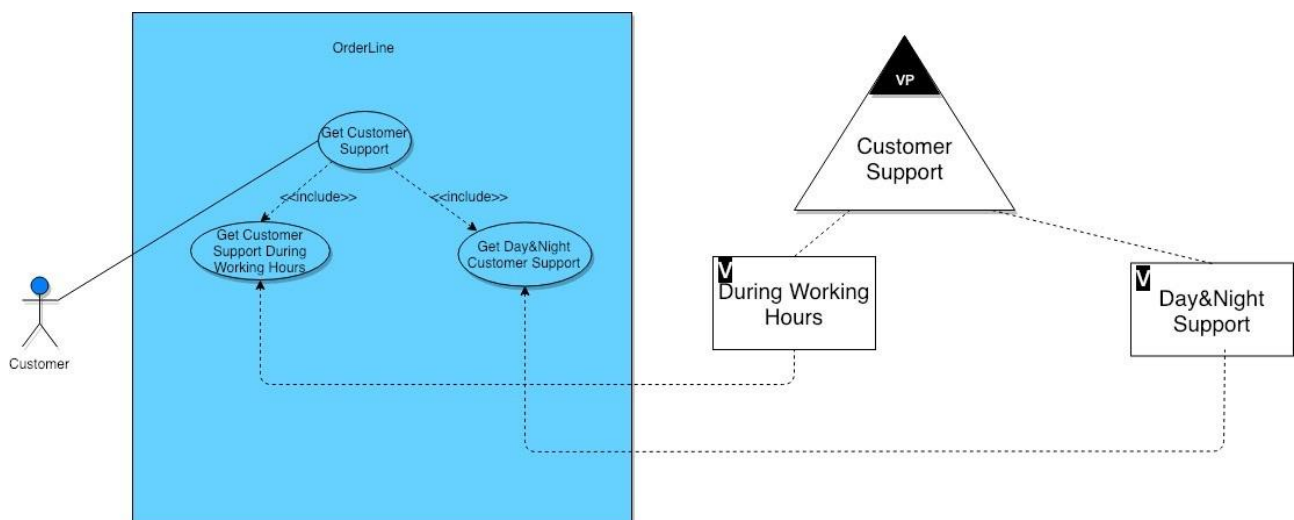




**Figure 14 :** Use case model for Collect Points

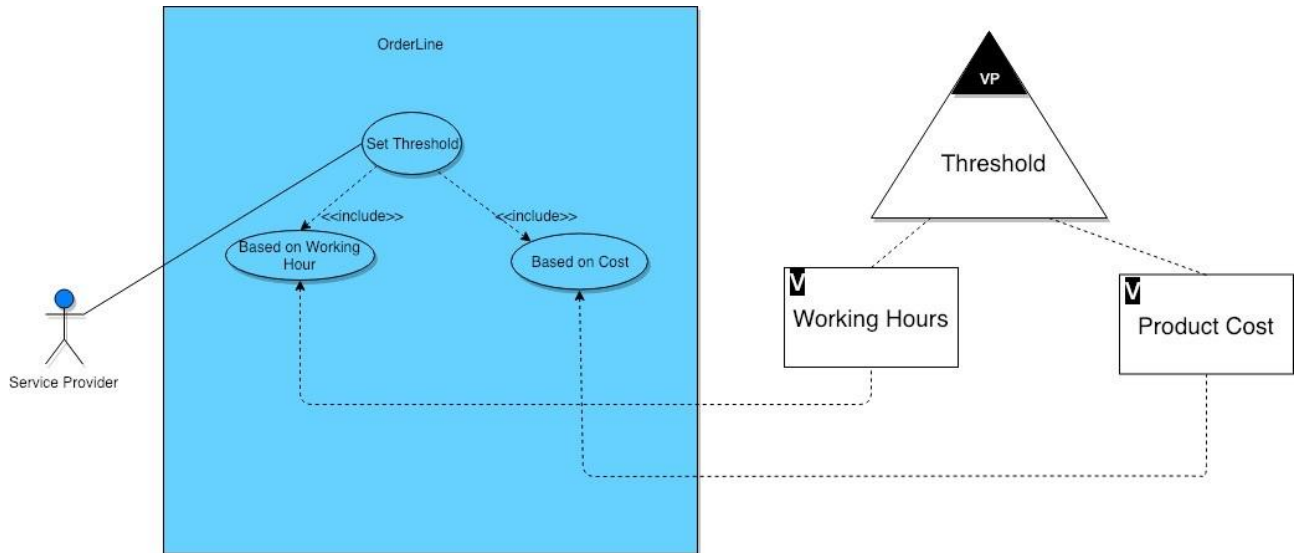
### 8.1.4 Customer Support

Users can get customer support in all applications. FeedMe, HealthyLife and MaturePear provide customer support during working hours. However, HealthyLife-Premium provides its premium subscribers 7/24 online customer support.



**Figure 15:** Use case model for Customer Support

## 8.1.5 Set Service Threshold



**Figure 16:** Use case model for Setting Service Threshold

## 8.2 Component Diagram

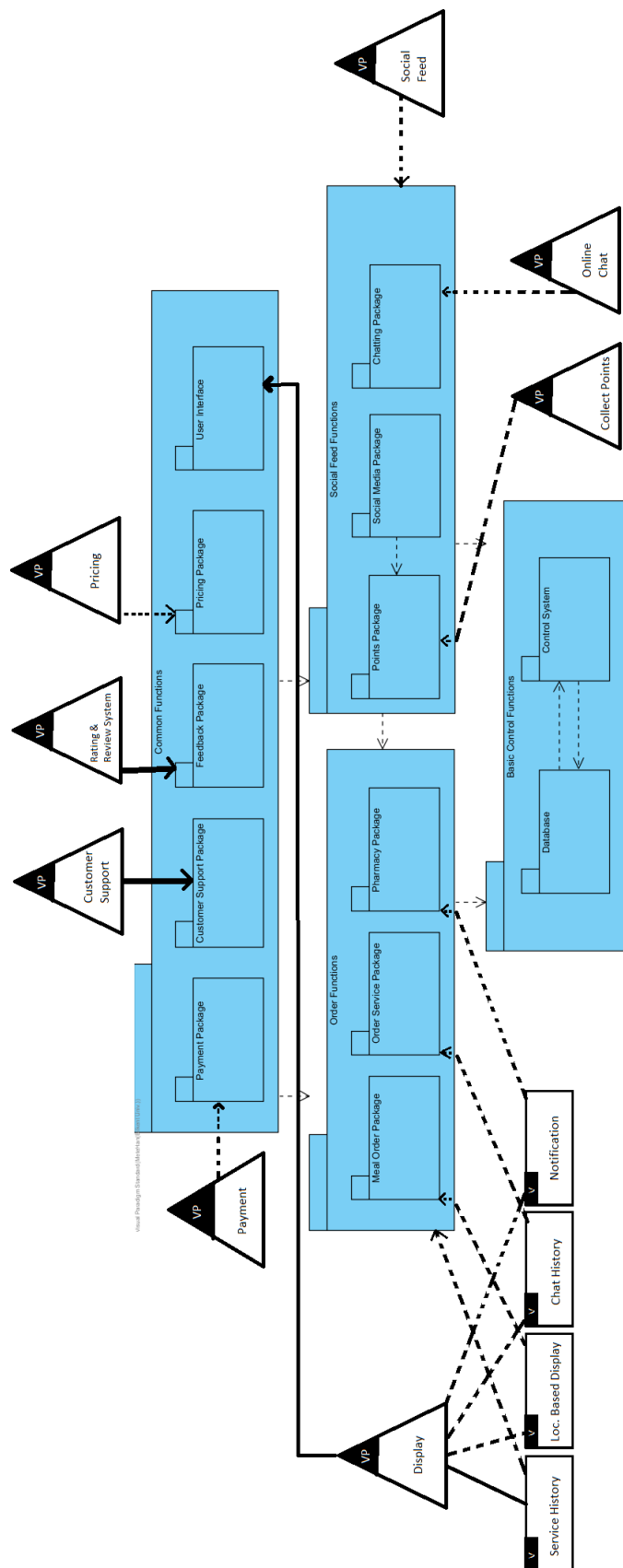


Figure 17 : Component Diagram of OrderLine

## 8.3 Class Diagram

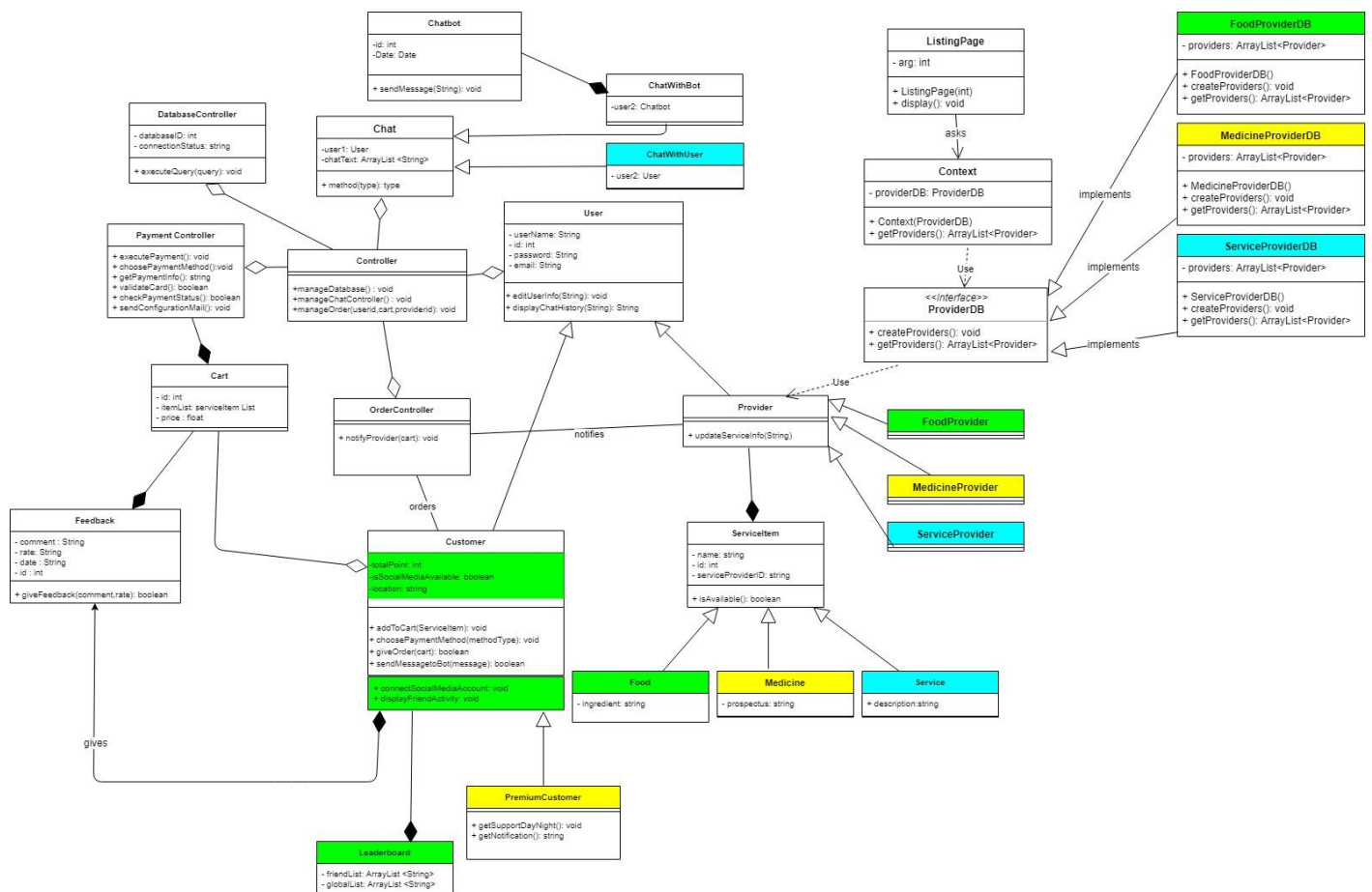


Figure 18 : Class Diagram of OrderLine

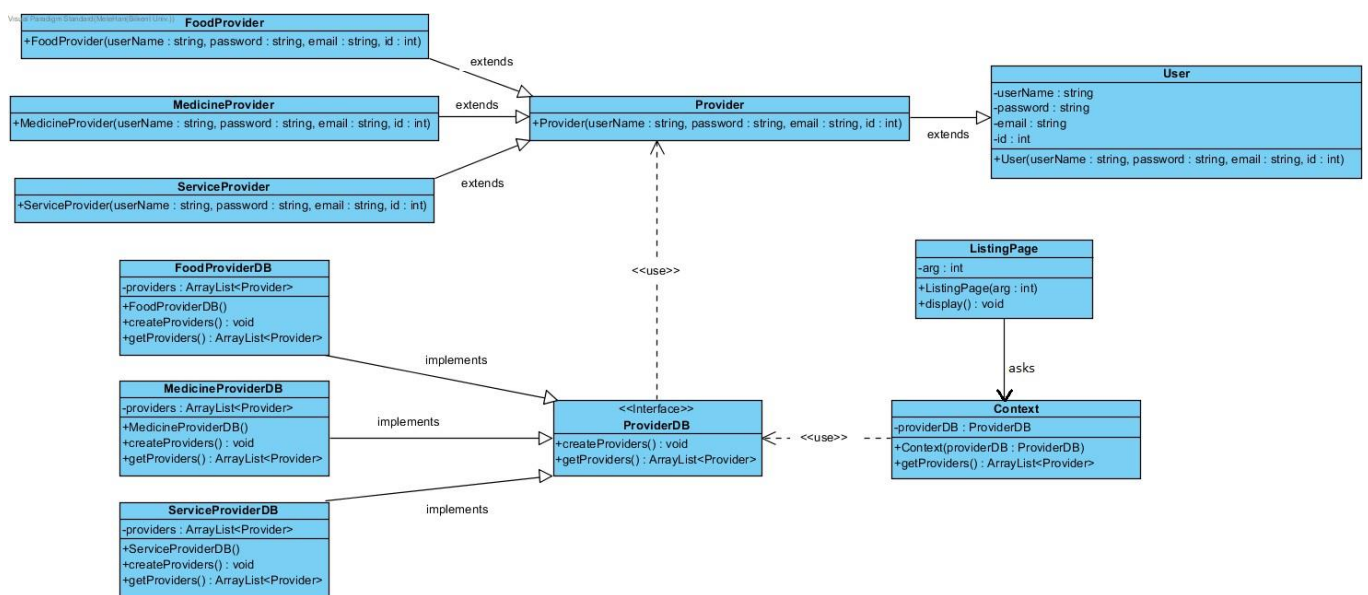
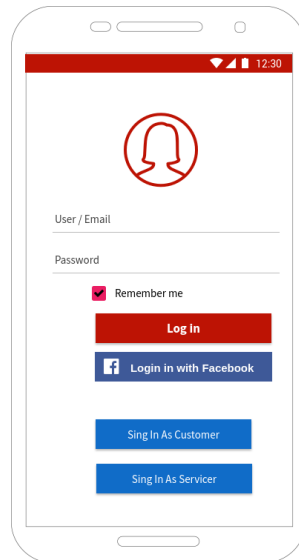


Figure 19: Strategy Design Pattern

## 9. Prototype

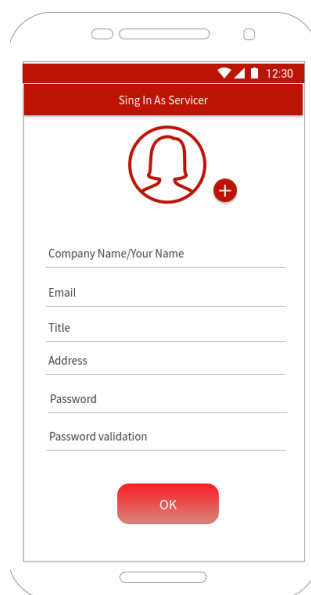
Some of mock-up design of three variants can be found below.

### 9.1 Common Prototypes



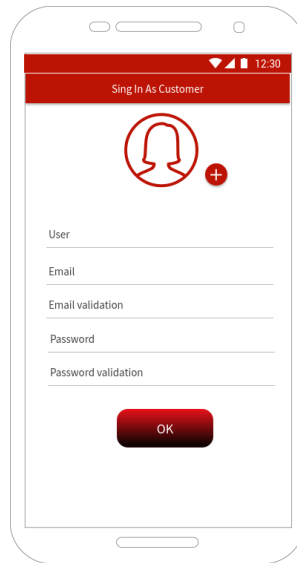
**Figure 20:** Login Page

When a user launches any of applications, she/he will be faced with an login page. They may log in with their username and password. Additionally, users may login with their facebook account in FeedMe to have a social network feed. If she/he does not have any account, she/he needs to sign in.



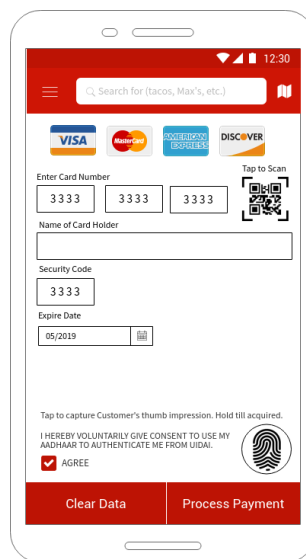
**Figure 21:** Sign in as servicer page

“Sign in as servicer” button is for registering the servicers for the applications. Such servicers are restaurants in FeedMe, architects, repair-mans, vs. in MaturePear, or pharmacies in HealthyLife.



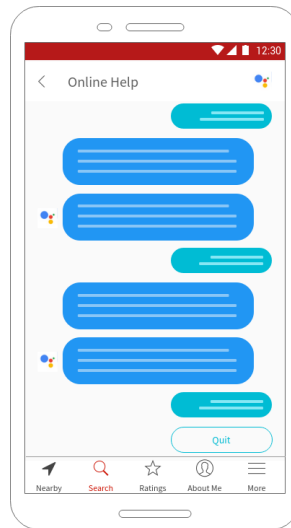
**Figure 22:** Sign in as customer page

This page is for customers to register that will use HealthyLife, MaturePear, and FeedMe.



**Figure 23:** Online payment

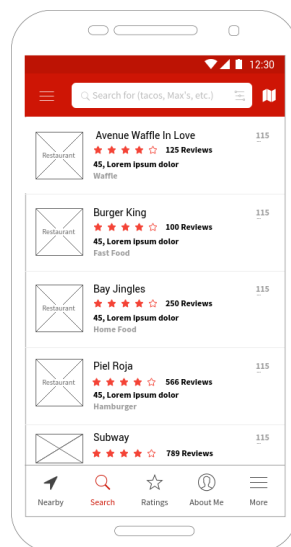
This page is optional for some applications since in FeedMe and MaturePear, users may make a payment cash.



**Figure 24:** Online help

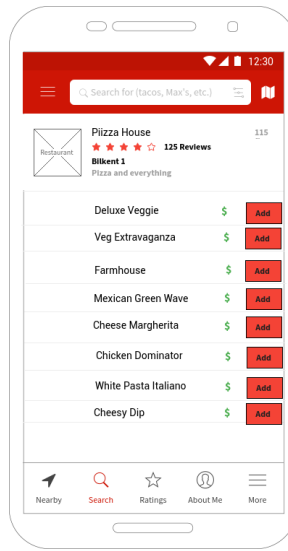
Online support is available for HealthyLife, MaturePear, and FeedMe.

## 9.2 Variable prototypes



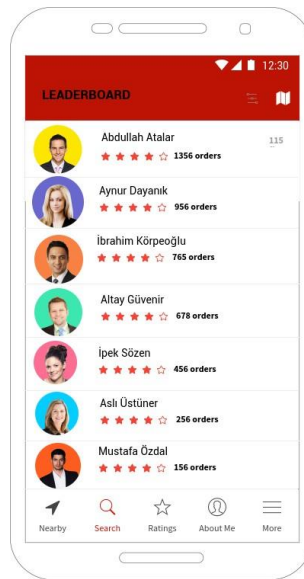
**Figure 25:** Listing all restaurants in FeedMe

All restaurants are listed respect to the user's location. The users may filter them by using filters on the top.



**Figure 26:** Listing all products in a restaurant in FeedMe

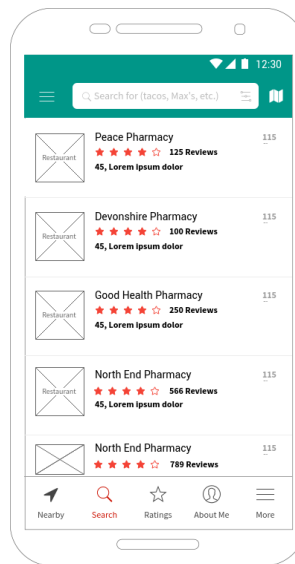
After listing all restaurants, the user may choose one of them, click on its image and all products served by the restaurant may be seen. The user may add these products to his/her cart.



**Figure 27:** Leadership board in the FeedMe

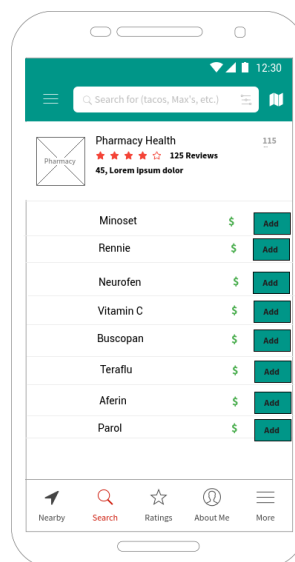
Leaderboard can be seen as list of the most active users in the application respect to their orders. This list consists of the user's facebook friends.





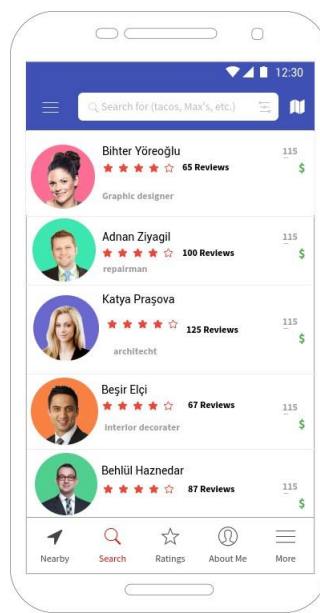
**Figure 28:** Listing all pharmacies in HealthyLife

All pharmacies are listed. Users may filter them by using filters on the top.



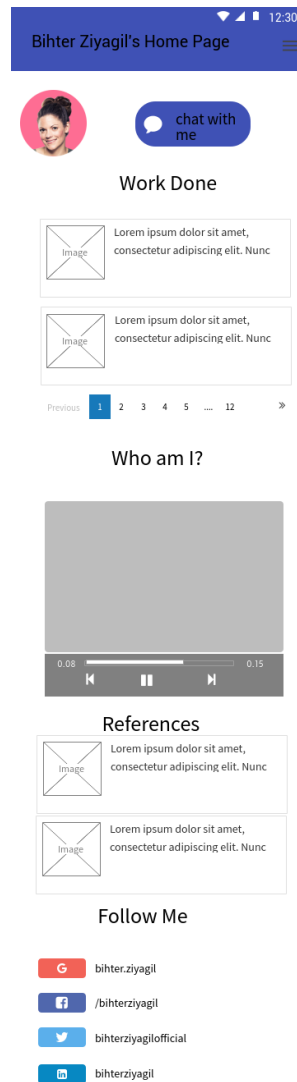
**Figure 29:** Listing all products in a pharmacy in HealthLife

After listing all pharmacies, the user may choose one of them, click on its image and all products served by the pharmacy may be seen. The user may add these products to his/her cart.



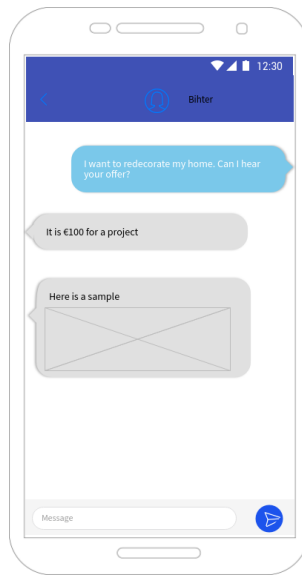
**Figure 30:** Listing service providers in MaturePear

Users may filter the list by using filters on the top. When a user clicks on name of a person, his/her customized personal page will be displayed.



**Figure 31:** Customized personal page the person who offer a service in MaturePear

Users may investigate servicer's works done, references, maybe watch her "who am I?" video to get to know her. If the user wants to get a service, she/he may start a chat with her.



**Figure 32:** Direct message with services in MaturePear

Users and service providers may directly chat and make a deal for a service.

### 9.3 Software Reuse in Prototype

All products of this project are different kinds of ordering systems. Therefore, it is possible to define put them together in the same software product line, similarity between them, and software reuse. Source code has some packages like CodeBase, GUI, FeedMe, MaturePear, HealthyLife where CodeBase and GUI are common. These common packages have common classes and some of them are associated with different FXML files like login, help, register, etc. For these files, it would be unreasonable to have different files for different products. For instance, HelpFormController.java and help.fxml are common since the logic of login is similar for these products. Login Panel has two users talking to each other, a customer and a provider. The provider sends a message, customer gets it, and replies it back. This logic is the same for all products.

```

Context context;

if( MainGUI.input == 1 ) { // FeedMe
    context = new Context( new FoodProviderDB() );
}
else if( MainGUI.input == 2 ) { // MaturePear
    context = new Context( new ServiceProviderDB() );
    paymentButton.setVisible(false);
    leaderboardButton.setVisible(false);
}
else { // HealthyLife
    context = new Context( new MedicineProviderDB() );
    leaderboardButton.setVisible(false);
}

ArrayList<Provider> providers = context.getProviders();
providerLabel1.setText( providers.get(0).getName() );
providerLabel2.setText( providers.get(1).getName() );
providerLabel3.setText( providers.get(2).getName() );

```

**Figure 33:** Software reuse for home page

Another example is home page of the apps. These pages are slightly different from each other. However, still it is possible to use a single java and fxml file for all products. According to value of the feature flag that corresponds to type of product, the buttons remain or not. Also, providers' information changes according to this flag. Here, also we use Strategy Design Pattern to get this information.

## 10. Discussion

Throughout the project, we have encountered some challenges which we overcome by seeking clarifications on Web and on books and also by reviewing topics that we have learned in this course. The challenges that we have met can be classified as follows:

### 10.1 Requirements

In order to make sure that we have a functional application, we first defined the generic requirements that our application is supposed to hold. These requirements consisted of primary functions which allow us to obtain an applicable product. Considering customers' potential needs, we have improved our requirements by adding some further features which may have an influence on customers' satisfaction in a positive manner. At this point, our main goal was to ensure that we have a broad set of requirements to evaluate on the Kano Methodology.

## 10.2 Feature Model

As known, feature model is crucial to show different kinds of product features that can be classified as common and variable. When we have completed our list of requirements for the system variants, we have proceeded to the next step which is to create the feature model. Yet, due to the fact that we have applied Kano methodology subsequently, we have realized that we were supposed to make some modifications on our requirements. This is the reason why we should have applied Kano methodology in the first place rather than forming the feature model. However, we have updated our feature model by making some corrections after the Kano methodology. Moreover, this project has differed from one of our assignments for which we have prepared a feature model based on provided functionalities to us but for our project, we were required to define all functionalities and to perform other phases of the project from scratch. Therefore, throughout the project, finding common and variable features as well as product specific features and creating the feature model of the system just according to this set of requirements were a challenge for us. But, there is no doubt that it was beneficial for us to figure out the ways that various kinds of features are represented.

## 10.3 Kano Methodology

During the project, we have defined feasible requirements according to our project description. Considering these requirements, we have prepared a set of questionnaire to be able to apply Kano methodology and this questionnaire was tested on all of the group members in order to obtain different point of views on the requirements. Yet, because of the fact that our questionnaire consisted of some requirements that have been decided in advance, we think that the primary purpose of the methodology could not be accomplished fully. But, other than that making a questionnaire from scratch was very helpful for us to learn the important points of it as we have talked about there might be some customer-oriented issues on the methodology including misunderstanding of questions and being bored of questions. This is the reason why we have paid attention to make simple and easy-to-understand questions to avoid losing the interest of the respondents. Furthermore, choosing right respondents from our target customers was another important point for the methodology. In spite of the fact that we have no real customers, as a respondent, we have tried our best to provide our answers from a customer point of view. Thus, we have gained experience about the overall phases of Kano methodology thanks to this project. As mentioned before, although we have some requirements decided in advance, applying this methodology was useful for us to properly identify commonalities and variabilities in the system and to eliminate some of our requirements.

## 10.4 Orthogonal Model

In order to show the variability in an explicit way, we have used orthogonal variability modeling. With the help of this model and use case diagrams, we were able to indicate points that variations have occurred. In other words, these have helped us to demonstrate what type of variation has occurred, how the variation has taken place and why we have needed these variations in our system. Furthermore, after having done some research, we have learned that there is a relation between SPL architecture and system behavior architecture. Namely, system functionalities and variabilities are affected from each other thus we have dependencies. This is the reason why we have chosen to use orthogonal variability model as it provides an easy-to-follow demonstration of variabilities. Moreover, this model was beneficial for us to explicitly see which set each component has belonged to. In addition, we will have the opportunity to identify components that may be influenced by dependencies if there is a modification on our application. As we have mentioned during the course, these relations are crucial to ensure consistency of information. Similarly, they are an essential part of the system for application requirement engineering.

Furthermore, due to the fact that we have considered our feature model to identify variation points and variants, we have first tried to organize systems that were related to each other and to show them using a single use case diagram. But, because of constraints, it has turned out that any error on notation could affect other variants. That's why we have separated these systems into many use cases.

Besides, there weren't any tool to create orthogonal model so we have had to use Visual Paradigm first to draw use case diagrams and then we have used draw.io to draw orthogonal model and to combine these two diagrams. It can be said that this operation was a waste of time for us.

## 10.5 Prototype

Considering that our application is a single product, we have designed it as it modifies its format depending on the chosen plan. That's why we have the opportunity to reuse UI components which are classified as common for every plan. Making this was beneficial for us to save time because we may choose to make distinct user views for each of them. But instead, we have used common components. Moreover, we are able to deactivate features which are plan-specific and which should not be present for other plans through checking a value which defines the chosen plan. In addition, in order to ensure that our functionalities work smoothly, we have made tests which operate in a unique manner for each plan.

## 11. Source Code

Source code of the project can be found in the following link: [GitHub-Repo](#)

Installing JavaFX packages may be necessary. Different products are available.

For different configurations follow the following instructions: Eclipse – Run Configurations – Arguments – Program Argument. Type 1 for FeedMe, 2 for MaturePear, 3 for HealthyLife. Then click Run.

## 12. Group Member Contribution

### **Berat Biçer**

He contributed on first version of Feature Model for Phase 1. He worked on designing some of the use case diagrams for Phase 1 Report. After Mr. Tüzün's feedback, he mainly worked on implementing better GUI for OrderLine. Also, he reviewed other team member's part in Phase 2 report and he gave feedback to his friends.

### **Ezgi Çakır**

She worked on some parts of reports. She prepared prototypes of applications for phase 1 report. She remade feature model diagram after Mr. Tüzün's feedback, then she and Seda went over it together and corrected. She mainly worked on implementation of OrderLine. Also, she worked on designing class diagram with Seda Gulkesen and Metehan Kaya. She helped Metehan Kaya to design the component diagram of OrderLine.

### **Gülce Karaçal**

She contributed to improve SPL idea and mainly worked on proposed variants and their explanations on the reports. She prepared some use cases for variant features of HealthyLife and FeedMe applications in Phase 1 Report. Finally, she prepared discussion parts for challenges we have faced throughout the project.

### **Metehan Kaya**

He created questionnaire for OrderLine and mainly contributed to Kano Methodology. He worked on designing class diagram with Ezgi Cakir and Seda Gulkesen. He worked on component diagram. He worked on implementation of Strategy Design Pattern. He helped finishing discussion part of

the final report. He added a missing panel, missing connections between panels and fixed minor GUI bugs.

### **Seda Gülkesen**

She contributed to writing commonalities, variabilities of the products. She explained all variants in detail. She contributed mainly on use case scenarios and she came up with Use Case Diagrams, feature model and class diagram. She worked on Feature Model with Ezgi Cakir. She collaboratively worked on designing class diagram with Ezgi Cakir and Metehan Kaya. The last version of the report is formatted by her. In addition to that, she prepared final presentation.