****

**Requirement**

**Specification**

**TEAM#4**

**한성주: 아이디어, system requirement,** **system architecture**

**김하영:system models**

**전재영 requirements specification**

**탕치 system evolution, appendices**

**로호연 introduction glossary**

**목차**

[1.PREFACE 4](#_Toc1273_WPSOffice_Level1)

[1.1 OBJECTIVES 4](#_Toc29083_WPSOffice_Level2)

[1.2 READERSHIP 5](#_Toc30111_WPSOffice_Level2)

[A. USER REQUIREMENTS READERSHIP 5](#_Toc29083_WPSOffice_Level3)

[B. SYSTEM REQUIREMENTS READERSHIP 5](#_Toc30111_WPSOffice_Level3)

[1.3 DOCUMENT STRUCTURE 6](#_Toc24534_WPSOffice_Level2)

[A. PREFACE 6](#_Toc24534_WPSOffice_Level3)

[B. INTRODUCTION 6](#_Toc9366_WPSOffice_Level3)

[C. GLOSSARY 6](#_Toc22937_WPSOffice_Level3)

[D. USER REQUIREMENTS DEFINITION 6](#_Toc18553_WPSOffice_Level3)

[E. SYSTEM ARCHITECTURE 7](#_Toc19212_WPSOffice_Level3)

[F. SYSTEM REQUIREMENTS SPECIFICATION 7](#_Toc10985_WPSOffice_Level3)

[G. SYSTEM MODELS 7](#_Toc5378_WPSOffice_Level3)

[H. SYSTEM EVOLUTION 7](#_Toc14423_WPSOffice_Level3)

[I. APPENDICES 7](#_Toc9366_WPSOffice_Level2)

[J. INDEX 8](#_Toc22937_WPSOffice_Level2)

[K. REFRENCE 8](#_Toc18553_WPSOffice_Level2)

[2. INTRODUCTION 8](#_Toc29083_WPSOffice_Level1)

[2.1 OBJECTIVES 8](#_Toc19212_WPSOffice_Level2)

[2.2 NEEDS 8](#_Toc10985_WPSOffice_Level2)

[Figure 1 Single family status and forecasts 9](#_Toc28008_WPSOffice_Level3)

[2.3 FOR SINGLE LIFE 12](#_Toc5378_WPSOffice_Level2)

[2.4 EXPECTED EFFECT 13](#_Toc14423_WPSOffice_Level2)

[A. for people who don't want to cook 13](#_Toc25913_WPSOffice_Level3)

[B. for Homemade kitchen 13](#_Toc7215_WPSOffice_Level3)

[3. GLOSSARY 14](#_Toc7064_WPSOffice_Level1)

[3.1 OBJECTIVES 14](#_Toc7064_WPSOffice_Level2)

[3.2 TERM DEFINITIONS, ACRONYM AND ABBREVIATION 14](#_Toc15208_WPSOffice_Level2)

[A. TERM DEFINITION 14](#_Toc7064_WPSOffice_Level3)

[A.1. User Terms 14](#_Toc15830_WPSOffice_Level2)

[A.2. Service Terms 14](#_Toc19681_WPSOffice_Level2)

[A.3. Develop computation-related terms 15](#_Toc26864_WPSOffice_Level2)

[4. User Requirements Definition 16](#_Toc15208_WPSOffice_Level1)

[4.1. Objective 16](#_Toc21666_WPSOffice_Level2)

[4.2. Functional Requirements 16](#_Toc16285_WPSOffice_Level2)

[A. Sign up 16](#_Toc15208_WPSOffice_Level3)

[B. Login 17](#_Toc15830_WPSOffice_Level3)

[C. Homemade food Sales registration 17](#_Toc19681_WPSOffice_Level3)

[D. food recommendation 17](#_Toc26864_WPSOffice_Level3)

[E. Recommendation evaluating 17](#_Toc21666_WPSOffice_Level3)

[F. Search food 17](#_Toc16285_WPSOffice_Level3)

[G. Homemade food purchase 18](#_Toc27497_WPSOffice_Level3)

[H. Reviews and ratings 18](#_Toc31998_WPSOffice_Level3)

[I. Recipe confirmation 18](#_Toc27497_WPSOffice_Level2)

[J. Community 18](#_Toc14275_WPSOffice_Level3)

[A.1. Performance Requirement 19](#_Toc31998_WPSOffice_Level2)

[A.2. Security Requirement 19](#_Toc14275_WPSOffice_Level2)

[A.3. Efficiency Requirement 19](#_Toc28052_WPSOffice_Level2)

[A.4. Dependability Requirement 20](#_Toc15284_WPSOffice_Level2)

[A.5. Usability Requirement 20](#_Toc23761_WPSOffice_Level2)

[B. Organization Requirement 20](#_Toc28052_WPSOffice_Level3)

[B.1. Environmental Requirement 20](#_Toc2161_WPSOffice_Level2)

[B.2. Operational Requirement 20](#_Toc1116_WPSOffice_Level2)

[B.3. Development Requirement 21](#_Toc20779_WPSOffice_Level2)

[C. External Requirement 21](#_Toc15284_WPSOffice_Level3)

[C.1. Regulatory Requirement 21](#_Toc12859_WPSOffice_Level2)

[C.2. Safety/Security Requirement 21](#_Toc23810_WPSOffice_Level2)

[5. System Architecture 23](#_Toc15830_WPSOffice_Level1)

[5.1. Objective 23](#_Toc17789_WPSOffice_Level2)

[5.2. ‘Log-in & Sign up’ System 23](#_Toc6535_WPSOffice_Level2)

[5.2. real time recommendation system 24](#_Toc23846_WPSOffice_Level2)

[5.3. purchase system 25](#_Toc23174_WPSOffice_Level2)

[5.4. community system 26](#_Toc17812_WPSOffice_Level2)

[6. System Requirement Specification 27](#_Toc17812_WPSOffice_Level1)

[6.1 Objective 27](#_Toc304_WPSOffice_Level2)

[6.2 Functional Requirements 27](#_Toc7125_WPSOffice_Level2)

[A. Sign up 28](#_Toc17812_WPSOffice_Level3)

[B. Login 28](#_Toc304_WPSOffice_Level3)

[C. Homemade food Sales registration 29](#_Toc7125_WPSOffice_Level3)

[D. Food Recommendation 29](#_Toc29995_WPSOffice_Level3)

[E. Recommendation Evaluating 30](#_Toc30791_WPSOffice_Level3)

[F. Search Food 30](#_Toc24778_WPSOffice_Level3)

[G. Homemade Food Purchase 30](#_Toc13358_WPSOffice_Level3)

[H. Reviews and ratings 31](#_Toc11240_WPSOffice_Level3)

[I. Recipe Confirmation 31](#_Toc29995_WPSOffice_Level2)

[J. Community 32](#_Toc25850_WPSOffice_Level3)

[7. System Models 40](#_Toc304_WPSOffice_Level1)

[7.1. Objective 40](#_Toc30791_WPSOffice_Level2)

[7.2. Context models 41](#_Toc28560_WPSOffice_Level2)

[A. Context Model 41](#_Toc28560_WPSOffice_Level3)

[B. Process Diagram 41](#_Toc18168_WPSOffice_Level3)

[B-1. Sign up process 41](#_Toc31188_WPSOffice_Level3)

[B-2. Selling process 42](#_Toc7542_WPSOffice_Level3)

[B-3. Purchasing process 43](#_Toc1949_WPSOffice_Level3)

[B-4. Community process 43](#_Toc27891_WPSOffice_Level3)

[7.3. Interaction models 44](#_Toc18168_WPSOffice_Level2)

[A. Use case Models 44](#_Toc10430_WPSOffice_Level3)

[B. Tubular description for each use case 45](#_Toc27752_WPSOffice_Level3)

[B-1. Join for User 45](#_Toc19702_WPSOffice_Level3)

[B-2. Login 46](#_Toc13980_WPSOffice_Level3)

[B-3. Purchase Food 46](#_Toc6541_WPSOffice_Level3)

[B-4. Food Upload 47](#_Toc3264_WPSOffice_Level3)

[B-5. Community 47](#_Toc22484_WPSOffice_Level3)

[B-6. List Near Seller’s Food 48](#_Toc17856_WPSOffice_Level3)

[C. Sequence Diagram 49](#_Toc5876_WPSOffice_Level3)

[C-1. Sequence Diagram of Purchasing Food 49](#_Toc31003_WPSOffice_Level3)

[C-2. Sequence Diagram of Log-in and Sign-up 50](#_Toc7503_WPSOffice_Level3)

[7.4. Structural models 51](#_Toc6541_WPSOffice_Level2)

[A. User System Class Diagram 51](#_Toc11496_WPSOffice_Level3)

[7.5. Behavioral Models 51](#_Toc3264_WPSOffice_Level2)

[A. Data-Driven Modelling 51](#_Toc17289_WPSOffice_Level3)

[A-1. User 52](#_Toc21128_WPSOffice_Level3)

[- Sign-up 52](#_Toc13779_WPSOffice_Level3)

[- Log-in 52](#_Toc28171_WPSOffice_Level3)

[- 개인정보 공유 53](#_Toc9803_WPSOffice_Level3)

[A-2. Purchasing User 53](#_Toc1348_WPSOffice_Level3)

[- 구매 가능 음식 목록 확인 53](#_Toc4576_WPSOffice_Level3)

[- 음식 상세정보 조회 54](#_Toc28007_WPSOffice_Level3)

[- 구매 요청 54](#_Toc28038_WPSOffice_Level2)

[A-3. Seller User 55](#_Toc6224_WPSOffice_Level2)

[- 음식 정보 업로드 55](#_Toc5074_WPSOffice_Level2)

[B. Event-Driven Modelling 56](#_Toc28038_WPSOffice_Level3)

[8.System Evolution 56](#_Toc28038_WPSOffice_Level1)

[8.1 Objective 57](#_Toc5718_WPSOffice_Level2)

[8.2 Limitation and Assumption 57](#_Toc13730_WPSOffice_Level2)

[8.3 Evolutions of Software 57](#_Toc4460_WPSOffice_Level2)

[A.Video Streaming 57](#_Toc6018_WPSOffice_Level3)

[C. Big Data Statistics 58](#_Toc24089_WPSOffice_Level3)

[8.4 Evolutions of User Requirement 58](#_Toc6018_WPSOffice_Level2)

[A. Business Needs 58](#_Toc29138_WPSOffice_Level3)

[B. Buyer's Demand 58](#_Toc16032_WPSOffice_Level3)

[8.5 Evolutions of Environment 58](#_Toc24089_WPSOffice_Level2)

[9.Appendices 59](#_Toc6018_WPSOffice_Level1)

[8.1 Objective 59](#_Toc16766_WPSOffice_Level2)

[9.2 Database requirements 59](#_Toc9185_WPSOffice_Level2)

[9.3 User-System Requirements 59](#_Toc20433_WPSOffice_Level2)

[9.4 Development process 60](#_Toc26347_WPSOffice_Level2)

[10.Index 61](#_Toc16766_WPSOffice_Level1)

[10.1 Figure Index 61](#_Toc21583_WPSOffice_Level2)

[10.2 Table Index 62](#_Toc10685_WPSOffice_Level2)

[11.Reference 65](#_Toc9185_WPSOffice_Level1)

**1.PREFACE**

**1.1 OBJECTIVES**

The intended audience, the overall demographic structure of the article, and the role of the parts are explained in the preface. And on the version management policy, version record, document changes and its basis.

* 1. **READERSHIP**

A detailed list of Requirements, depending on the reader, generally consists of two parts, User Requirements and System Requirements. The User Requirements are the requirements from the user's point of view, allowing the developer to understand the user's original requirements, and the System Requirements are the ones that detail the requirements in detail.

1. **USER REQUIREMENTS READERSHIP**

After we have identified which users, we will analyze the specific user needs. On our software level, we need to analyze the different needs of ordering and selling.

User demand is the most difficult requirement mining in the process of product execution. Should follow the user or guide the user, how to deeply dig the user needs, how to transform the user needs into product needs

1. **SYSTEM REQUIREMENTS READERSHIP**

System requirements are technical requirements completely hidden in the implementation of technology. Such as user login function, user account name length character requirements, whether to encrypt transmission. Whether you want to use a distributed server, whether you want to do redis caching, whether you want to use a distributed database, etc. However, with the launch of the product, the internal operation and related personnel will also propose to develop an operation system and how to treat the product requirements at this time. In fact, at this time, the operation management system is the business demand, the operator is converted to the user, he put forward the demand is the user demand.

* 1. **DOCUMENT STRUCTURE**

This document consists of preface, introduction, glossary, user requirements definition, system architecture system requirements specification, system models, system evolution, appendices, index 10 parts.

1. **PREFACE**

Preface explains the intended audience for this article, the overall layout of the article, and the role and content of each section. Put forward the version management policy, and record the version change according to this table, and describe the additional document change items and basis.

1. **INTRODUCTION**

In Introduction, the necessity of "For Single Life" and what requirements "For Single Life" reflects. And a summary of how the system's functions and other functions function. It also explains how the system achieves the goal.

1. **GLOSSARY**

The use of technical terms in the document is described in the glossary part.

1. **USER REQUIREMENTS DEFINITION**

This section describes the services that must be provided by the system and the constraints that products or system procedures need to comply with.

1. **SYSTEM ARCHITECTURE**

System Architecture provides a high-level overview of the target system, as well as an overall distribution of system functionality

1. **SYSTEM REQUIREMENTS SPECIFICATION**

The System Requirements Specification further clarifies functional and non-functional requirements in more detail in the form of requirements for most functions that are further subdivided. Requirements specification in order to record the requirements analysis of this software design is the final result, as well as the data and functions that the software designer will use in the future. The reader is a member of the team that writes the software.

1. **SYSTEM MODELS**

System Models describe the components of the System and how the System interacts when using specific capabilities

1. **SYSTEM EVOLUTION**

All possible changes to the target system in the future are also described. Such as the development of the system, changes in user requirements, etc.

1. **APPENDICES**

Minimum system setup required to use the system, as well as database structure and development process.

1. **INDEX**

Index collates the Index of the file. Includes caption picture index, table index, etc.

**K. REFRENCE**

Declare this system development reference documentation and source code

**2. INTRODUCTION**

**2.1 OBJECTIVES**

Explain the necessity of FSL system in social demand and market development, and describe the corresponding demand of the system. In addition, a brief description of the purpose of the system how to achieve, and the desired results.

**2.2 NEEDS**

More and more people are living alone these days. With the increasing number of college students leaving their homes to go to or from the capital for college, the rate of self-employment has increased significantly. Not only college students, but also many people take the exam for employment. Many people live alone from other places. Besides, because the distance from home to work is too far, in order to make commuting easier, they live alone in the workplace. Like this, the number of renters living alone at home is increasing because of the location of colleges, job preparation, and corporate commuting.

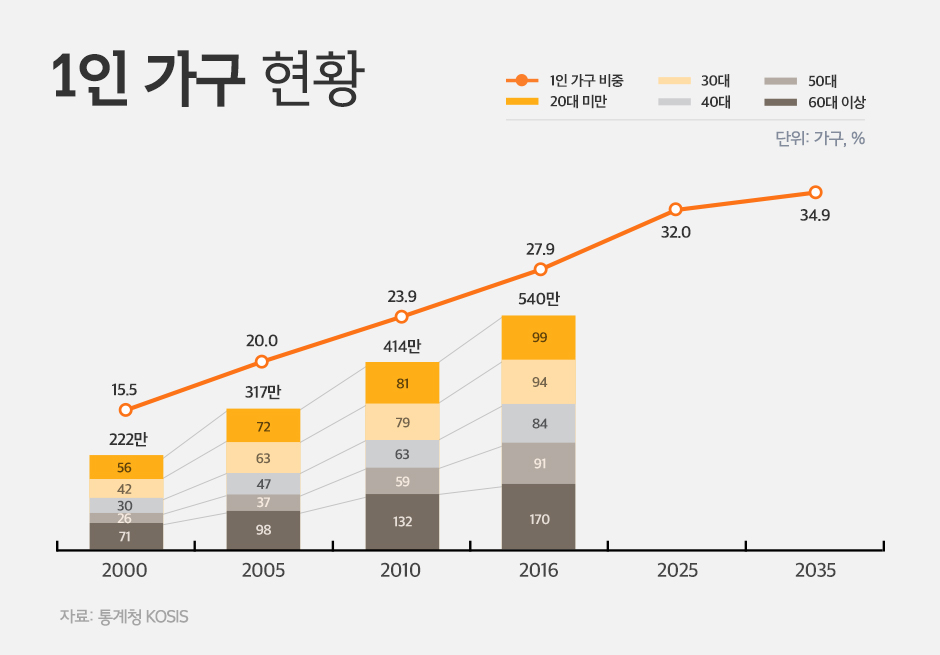
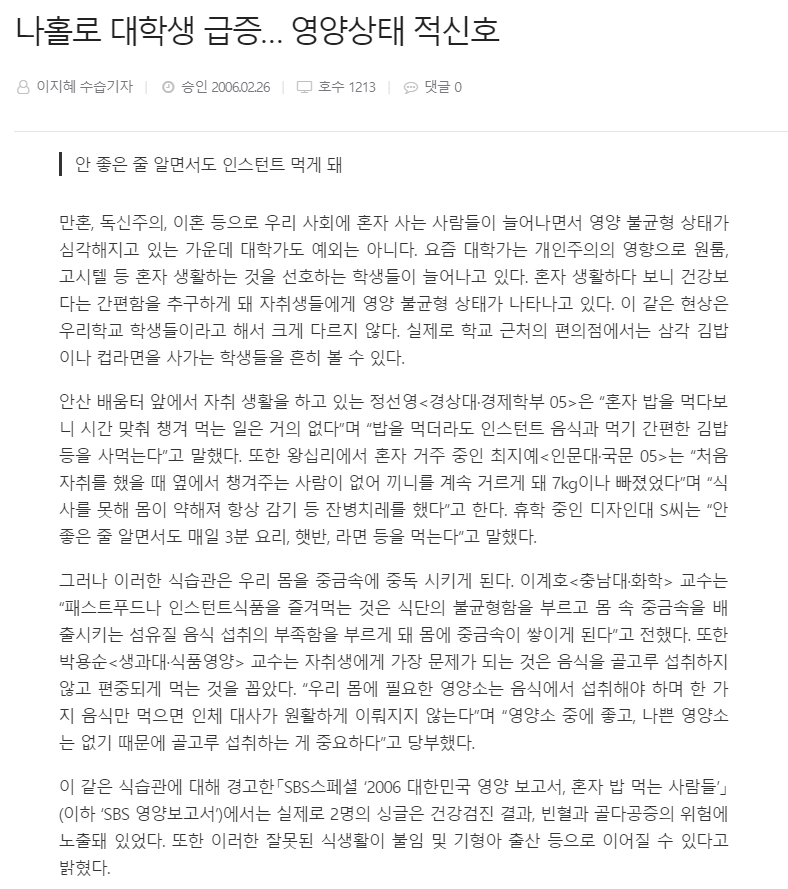


Figure 1 Single family status and forecasts

According to the general population survey conducted by the census and statistics department in 2016, there were about 5.4 million households with one person in 2016, accounting for 27.9% of the total number of households, an increase of about 4 percentage points over 2010.In addition, the census and statistics department projects that by 2035, one household will account for 34.9 per cent of all households. That is to say, in the near future, one third of the families will become one person family, with one person family as the center, and the trend and lifestyle will change.

Although the number of one-person families has increased dramatically, there are many problems in life as a one-person family .In 1 family, especially in the college students who are learning self-cooking, the largest proportion is "diet" and "diet" problem. People who live alone are not used to cooking. If they cook by themselves, it is difficult to adjust the quantity and deal with the rest of the food. Cooking is troublesome and it is difficult to do anything by themselves.



**Figure 2 The problem of malnutrition among those living alone in news**



**Figure 3 The problem of malnutrition among those living alone in internet searching**

The biggest problem caused by this self-feeding mistake is unbalanced nutrition intake. Most fast food has many irritants, salts, animal proteins and fats. On the contrary, due to the lack of vitamins and inorganic substances, it is easy to cause nutritional imbalance, which will damage the nutritional balance of the human body and significantly reduce the immune function. These health problems are likely to occur because people who cook for themselves often eat ready-made food. A search on the web portal for "nutrition issues in self-cooking" reveals many people who are experiencing nutritional imbalances. Among them, some students began to cook their own life, and eat a lot of processed food, so often get enteritis, some students said that exercise often appear dizziness and vomiting and other symptoms.

In addition, fast food is rich in protein, fat, sugar will also become the cause of obesity. In particular, fast food does contain a lot of sugar, which can lead to fatigue, headache, inattention, restlessness, anxiety and other symptoms. This can lead to more fast food being found, which can lead to a vicious cycle in which sugar must be continuously consumed, leading to diabetes.

**2.3 FOR SINGLE LIFE**

Seeing the problems of self-cooking students, we will think that as long as the website that recommends and shares recipes for self-cooking students can provide "self-cooking food ordering service", the burden of self-cooking students cooking at home can be slightly overcome .With this in mind, the author thinks of the service of selling and buying food among self-eaters on the basis of sharing existing recipes, namely real-time ordering service .After using this function, those who can't cook at all or have no materials at home, even those who dare not to cook because of food waste can easily eat home-cooked meals, thus providing a balanced intake of nutrients.

Based on the instant ordering system and location service, we have set a goal to provide more functions on the existing recipe website.

Based on the above requirements, we have developed the basic functions of the system, and the following functions will be added or modified according to the development of the software

**2.3.1. Real-time information sharing function**

When you sell food, you can generate posts, upload pictures of the food and video and text. In this way, people who buy food can easily get information about the food.

**2.3.2. Short time seller information providing function**

Since when buying food for cooking, on the basis of location service, provide the recent sale of food within a few minutes and at least a few meters of the seller information and find the function of the homemade food made of kitchen bulletin, provide direct trading for the nearest solitary, convenient food shopping service.

**2.3.3. Real-time dining notification function**

When you buy the food, you should contact the seller's contact information, you can directly contact SMS or SNS instant messaging tools. This way the seller can confirm the arrival of the notice as soon as possible and contact the buyer to provide the food.

**2.4 EXPECTED EFFECT**

**A. for people who don't want to cook**

1. Eat foods that are healthier and more balanced than convenience store fast food
2. No more kitchen waste from cooking by yourself.
3. It also saves time on cooking.
4. Saves time cleaning the kitchen.
5. You can eat more flavors.
6. Saves a lot of money on takeout orders.
7. More convenient know yourself what homemade around the kitchen.

**B. for Homemade kitchen**

1. The amount of material used in cooking does not require careful control
2. A big meal does not go to waste
3. Earn change by selling through this system
4. Improve cooking skills through user feedback
5. A platform for people who love to cook
6. **GLOSSARY** 
   1. **OBJECTIVES**

Glossary explains the use of technical terms in the document.

**3.2 TERM DEFINITIONS, ACRONYM AND ABBREVIATION**

**A. TERM DEFINITION**

**A.1. User Terms**

|  |  |
| --- | --- |
| Term | Definition |
| Users | All user definitions for the FSL system |
| Administer | The administrator of this service system is responsible for the operation and maintenance of the server |
| Homemade Kitchen | The name of the private chef in the system, in this platform to sell their own cooking |
| Buyers | Buyers of meals cooked by others on this platform |

**Figure 4 User Terms**

**A.2. Service Terms**

|  |  |
| --- | --- |
| Terms | Definition |
| Login | To use the service, log in by comparing the previously registered information with your own information to see the previously reserved information. |
| Sign up | The new user transmits the user information to the platform to apply for a user account |
| Recommended | Recommend nearby kitchenware to buyers |
| Place the order | Buying food |
| Upload the menu | Upload food information, including pictures, names, descriptions, video |
| Accept orders | The acceptance of an order service indicates that a contract has been reached between the buyer and the seller |
| Cancel orders | Cancel the order and explain why |
| Near | Check out the nearby kitchen |

**Figure 5 Service Terms**

**A.3. Develop computation-related terms**

|  |  |
| --- | --- |
| Terms | Definition |
| DB | Data Base to store the user information |
| Server | Compute user requests and store DB |
| Pay | Payment is made through an external payment system |

**Figure 6 Develop computation-related terms**

**4. User Requirements Definition**

**4.1. Objective**

User Requirements Definition 에서는 사용자 즉 개개인의 요리 구매자와 판매자에게 제공되는 서비스테 대해 설명한다. 내용은 자연어와 다이어그램과 같은 전달방법을 사용하여 독자가 이해하기 편하도록 합니다. 내용으로는 제품의 시스템 프로세스가 따라야 할 제약사항과 Functional Requirements를 포함한 Non-Functional Requirements에 대해서 상세히 기술한다.

**4.2. Functional Requirements**

For single life에서 제공될 기능은 아래와 같다. 시스템은 하단에 서술된 요구사항들을 충족한다.

1. **Sign up**

Sign up은 회원가입을 위해 사용자의 개인정보를 요청하고 사용자에게 제공받은 개인정보를 저장하는 기능이다. 사용자는 For single life가 제공하는 서비스를 이용하기 위해서는 sign up을 진행해야 한다. 개인간의 거래를 추구하는 서비스 이기 때문에 회원가입에서는 구매자와 판매자에 대해 구분하지는 않는다. 회원가입의 편리성을 위해 소셜 네트워크와 연동하여 회원 가입을 진행할 수 있는 기능을 제공한다.

모든 회원가입 과정에서는 기본적으로 아이디, 비밀번호, 성명, 성별, 전화번호 그리고 주소를 기본 정보로 입력하게 되며 전화번호를 통해 본인 인증을 진행하게 된다. 모든 정보가 정확하게 입력되고 전화를 통해 인증이 완료될 경우 회원가입이 완료되고 계정 정보가 생성되며 정보들은 user DB에 저장된다.

1. **Login**

사용자가 회원가입을 통해 생성한 아이디와 비밀번호를 사용하여 시스템에 로그인하는 기능이다. 사용자는 가입된 아이디를 사용하여 시스템에 로그인하여 자신의 데이터베이스에 접근할 수 있으며 For single life가 사용자의 정보를 바탕으로 서비스를 제공하게 됩니다.

1. **Homemade food Sales registration**

For single life가 제공하는 기본적인 기능으로, 일상생활속에서 개개인의 사용자가 자신이 만들어 놓은 요리에 대해 판매를 하기 위해 등록을 하는 기능이다. 판매할 제품을 등록을 할 경우에는 연락을 하기 위한 전화번호를 연동을 할 필요가 있다. 등록된 요리는 일정 거리 내에 있는 사용자와 연결시킴으로써 거래가 가능하게 한다.

1. **food recommendation**

사용자가 로그인을 진행한 후에 기본적으로 바로 사용이 가능한 기능이다. 개개인이 거래 가능한 거리내에 있는 개인 판매자가 등록한 판매 상품에 대해서만 추천을 한다. 추천의 우선순위를 결정하기 위해서 판매자의 평가나 현재 판매중인 상품의 평가 등이 고려될 것이다.

1. **Recommendation evaluating**

사용자가 추천 받은 음식에 대한 피드백을 제공할 수 있는 기능이다. 사용자가 마음에 들지 않는 음식을 추천화면에서 삭제할 수 있다.

1. **Search food**

사용자가 추천 받은 음식 이외에 먹고 싶은 메뉴에 대해 키워드로 자신이 원하는 음식에 대해 검색을 하는 기능이다. 음식의 카테고리별로 다양한 필터를 적용하면 보다 편리한 검색기능이 완성 될 것이다.

1. **Homemade food purchase**

사용자가 자산이 원하는 음식을 구입하는 기능이다. 원하는 음식이 존재할 경우 해당 음식을 등록한 판매자와 거래를 진행 할 수 있다. 구입을 결정하게 되면 등록된 판매자의 전화로 연락이 가게 되고 판매자가 구매에 대해 확인을 하게 되면 거래가 성사된다.

1. **Reviews and ratings**

사용자가 음식을 구입한 후 해당 음식과 음식을 등록한 판매자에 대해 평가를 할 수 있다. 이 평가를 통해서 사용자들은 판매자의 신뢰성을 평가 할 수 있을 것이다. 또한 추천음식의 제시 기준 또한 이러한 평가를 기준으로 제공 될 것이다.

1. **Recipe confirmation**

사용자가 다른 사용자가 등록한 음식의 레시피를 확인 할 수 있는 기능이다. 등록된 레시피는 유료/무료에 따라 결제 서비스를 포함하게 된다.

1. **Community**

사용자들 간의 소통을 제공하는 기능이다. 사용자 간의 공감대를 형성하고 서로의 정보를 나눌 수 있는 자유로운 공간이다. 작성된 게시물은 날짜와 시간순서에 따라 최신의 글이 상위에 노출된다. 댓글을 작성할 수 있으면 검색 기능을 사용해 과거의 게시물을 찾아볼 수도 있다.

**4.2. Functional Requirements**

Non-Functional Requirement는 Product, Organization, External Requirement로 나누어 각 요소별로 서술될 것이다. 그리고 각 요소 마다 세부적으로 더 분류 도리 수 있지만 여기서는 본 서비스에서 필요 되는 것만 서술하고자 한다.

1. **Product Requirement**

**A.1. Performance Requirement**

해당 시스템은 실시간 개인간 거래 서비스이므로 유연한 서비스를 제공해야한다. 거래중의 과정들이 바로 확인이 가능해야 하며 내용의 누락이 없어야 하기 때문에 performance적인 측면에서 문제가 없도록 해야 한다. 또한 사용자의 판매등록이나 등록한 음식에 대한 추천 필터 적용 등에 시간적인 문제가 발생 해서는 안되고 불필요한 지연에 인해 사용자가 불편함을 겪는 일이 없어야 한다.

**A.2. Security Requirement**

해당 시스템의 개인간의 거래를 추구하기 위해 사용자의 주소나 전화번호와 같은 중요한 개인정보를 수입하기 때문에 외부의 악의적인 침입으로부터 해당 시스템을 보호하고, 고객의 개인정보가 유출되는 불상사가 없도록 보안성이 뛰어난 시스템을 사용한다. 또한 관리자를 위한 접속 경로는 관리자를 제외한 다른 사용자들의 접근을 방지하기 위해 관리자만이 알 수 있도록 설계되어야 한다. 관리자를 제외하고는 사용자만이 고유의 아이디/비밀번호 등을 이용하여 자신의 데이터베이스에 접근하여 관리할 수 있다.

**A.3. Efficiency Requirement**

해당 시스템은 서버와 지속적인 통신을 전제하여야 한다. 실시간 상품 추천과 거래를 지원하는 시스템이기 때문에 거래진행을 할 때 정보를 불러오는 대기 시간이 길어 지지 않도록 해야 한다. 서버와의 지속적인 통신의 전제는 매우 중요한 요소로 작용된다. 따라서, 서버와 주고 받는 데이터를 최소화하야 하며 통신의 전환에 소요되는 시간 또한 최소화 해야 한다.

**A.4. Dependability Requirement**

해당 시스템은 가까운 거리의 고객 간의 거래를 가능하게 하기 위해서 실시간으로 거래가 진행되기 때문에 거래중인 상품이 임의로 변경되거나 삭제 되지 않고 유지 되어야한다. 또한 판매자가 등록상품에 대해 수정 및 변경 또는 삭제를 할 경우 즉각적으로 반영을 하도록 해야 한다. 따라서 안정적인 서버를 구축하여 문제가 일어나지 않도록 사용자에게 신뢰도가 높은 서비스를 제공 해야 한다.

**A.5. Usability Requirement**

해당 시스템은 사용자의 사이트 이용에 도움을 줄 수 있는 기능을 제공해야 한다. 카테고리를 설정하여 사용자들이 자신이 원하는 음식을 쉽고 간편하게 검색할 수 있도록 한다. 사용자들이 직관적으로 이해 할 수 있는 버튼의 배치 등 좋은 UI/UX를 제공해야 한다. 또한 자신의 거래중인 상품이나 거래 완료된 상품에 대해 확인 할 수 있는 페이지를 제공하고, 거래중의 알람 서비스를 제공한다.

1. **Organization Requirement**

**B.1. Environmental Requirement**

해당 시스템은 부트스트랩을 사용한 반응형 웹기반으로 개발되어, 사용자의 환경에 최적화된 화면을 제공한다. 반응형 웹의 경우 지원하지 않는 브라우저의 버전도 있으므로, 이에 대해 사용자에게 명시를 하거나 이에 대한 대비책을 준비해야 한다.

**B.2. Operational Requirement**

해당 시스템은 개인간 실시간 거래 서비스 이므로 판매자와 구매자가 거래를 진행을 할 때 서로 간의 채팅과 같은 실시간 커뮤니케이션이 원활해야 하고 구매자와 판매자사이의 통신을 지원한다. 사용자들은 커뮤니티에서 게시글을 자성함으로써 정보를 공유 할 수 있고 레시피를 유료/무료로 이용함으로써 정보를 얻을 수도 있다. 음식의 구입을 마치고 각각의 음식과 판매자에 대해 평가를 할 수 있으며 또한 추천 받은 음식이 마음에 들지 않는 경우에 삭제하는 기능을 제공한다.

**B.3. Development Requirement**

개인간의 실시간거래를 제공해야 하기 때문에 빅데이터와 머신 러닝을 사용하여 일정 거리내의 사용자가 등록한 음식만이 구매자에게 보이도록 구성을 해야 한다. 또한 단말기의 종류에 영향을 덜 받는 웹 서비스를 채택하고, Django 프레임워크를 통해 웹 서비스 간의 호환성을 높인다.

1. **External Requirement**

**C.1. Regulatory Requirement**

해당 시스템은 사용자의 개인정보를 입력 받는 회원 가입을 해야 서비스를 이용할 수 있는 시스템이다. 해당 시스템은 사용자의 개인정보(거래/결제 정보 포함)를 외부에 제공할 경우 사용자의 동의를 우 선으로 받아야 한다. 사용자가 거래 및 결제, 그리고 추천 시스템을 이용하는데 필수적인 정보의 경우에는 회원가입 과정에서 이용약관에 명시하여 동의하도록 한다. 그 외의 경우에는 사용자의 개인정보에 접근할 수 없도록 해야 한다.

**C.2. Safety/Security Requirement**

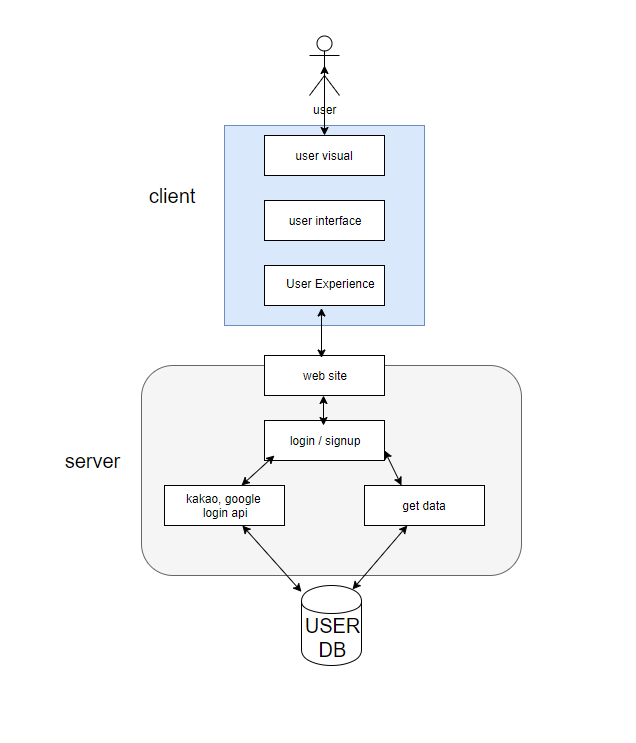
모든 처리가 중앙 서버를 한 번 이상 경유해야 하므로 중앙 서버가 네트워크 과부하에 의해 오작동하지 않도록 서버 용량 및 트래픽에 대한 예측이 빗나가지 않도록 해야 한다. 또한 해당 시스템은 고객의 개인 정보가 외부의 접근으로부터 안전하게 여러 방법을 통해 보안성을 높여야 한다. 서버 관리자는 해당 서비스에 회원 가입하여 시스템을 이용하는 사용자들을 관리하고, 악의적인 접근을 막아야한다.

**5. System Architecture**

**5.1. Objective**

System architecture에서는 본 시스템을 구성하는 architecture 개요를 고수준으로 설명합니다. 시스템을 구성하는 컴포넌트를 명시하고, 이들 간의 상호작용을 시각화를 통해 그림으로 제시함으로써 시스템 기능의 전체적 분포를 보여준다.

**5.2. ‘****Log-in & Sign up’ System**



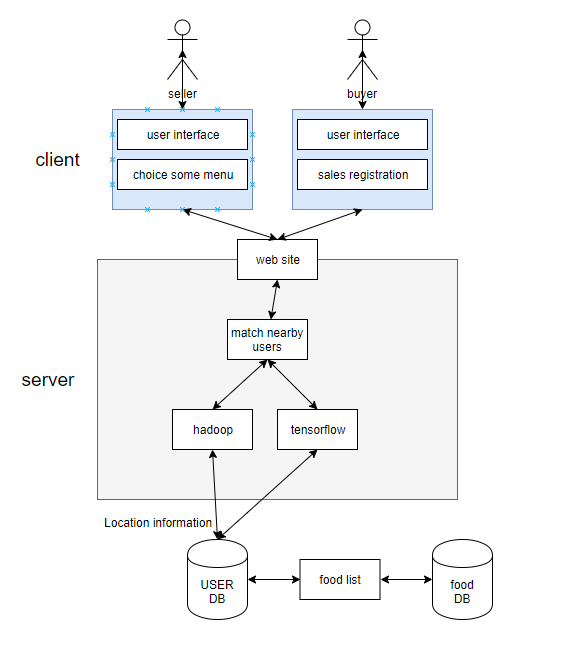
**Figure 4 Log-in & Sign up’ System**

‘Sign in & Sign up’ System은 user가 For single life에 접속했을 때, log-in이나 ‘sign up’을 할 수 있도록 하는 도와주는 시스템이다

user는 구글이나 카카오톡 네이버를 통한 로그인을 이용하거나 For single life에 sign up을 할 수 있다. 이때, system은 ‘User Database’에서 혹시 가입이력이 있는 지를 확인한다. 가입이력이 있다면 user에게 알리고 log-in하도록 한다. 가입이력이 없다면 sign up을 계속하도록 한다. Sign up을 하는 중에는 user 가 사용하려는 아이디가 다른 유저의 것과 중복되는 지를 확인한다. 중복된다면 user에게 알려 다른 아이디를 사용하도록 하고, 중복되지 않는다면 sign up을 마무리하도록 한다. Sign up을 할 때 user가 입력한 정보들은 모두 ‘User Database’에 저장된다.

User가 log-in을 할 때는, user가 입력한 아이디가 ‘User Database’에 존재하는지, 존재한다면 아이디에 해당하는 비밀번호가 user가 입력한 비밀번호와 일치하는 지를 확인한다. 이 세가 지 경우들 중에서 하나라도 틀렸다면 user에게 알리고 재입력하도록 유도한다.

**5.2. real time recommendation system**

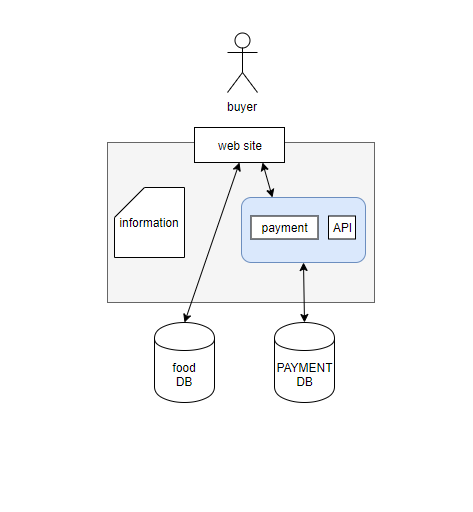


**Figure 5 real time recommendation system**

사용자들이 개인적으로 거래를 진행하기 위해 실시간으로 판매되고 있는 음식에 대해 추천을 받는 시스템이다.

이 시스템은 가까운 거리의 개인를 연결 시켜 거래가 이루어 지도록 해야 하기 때문에 user database에서 사용자의 주소를 기본정보로 판매자가 상품을 등록 했을 경우 hadoop 또는 TensorFlow를 사용하여 그와 가까운 거리의 사용자의 추천화면 또는 검색화면에 나타나도록 한다.

**5.3.** **purchase system**

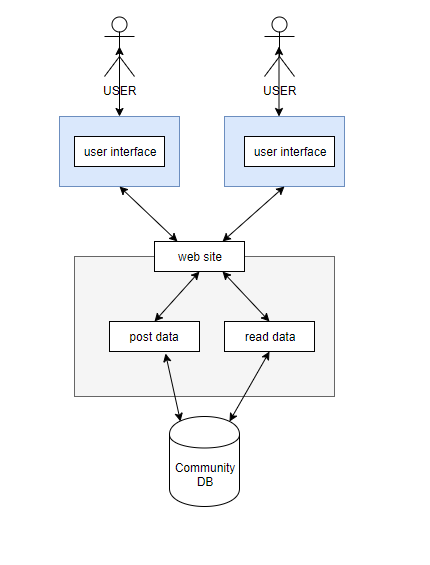


**Figure 6 purchase system**

사용자가 음식을 구매하기위한 시스템이다. 원하는 음식이 있을 경우에 해당 음식을 주문할 수 있다.이때 결제를 하기 위해서 결제서비스를 포함하게 된다.

**5.4.** **community system**

커뮤니티 시스템은 공유 된 을 게시판 형식으로 정렬하여 모든 사용자가 읽을 수 있도록 제공하는 시스템이다. 이 시스템에서 사용자들은 자유롭게 정보를 공유할 수 있다. 또한 댓글 기능이 있어 글작성자에게 피드백을 줄 수 있다.



**Diagram 7 community system**

**6. System Requirement Specification**

**6.1 Objective**

이 장에서는 기능적 요구사항과 비기능적 요구사항에 대해 자세하게 설명하고, 시스템을 이용하는 것에 대한 시나리오를 다룬다.

**6.2 Functional Requirements**

**A. Sign up**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 회원 가입 |
| Explanation | 사용자에게 회원 가입을 위해 필요한 정보를 요청하고 사용자에게 제공받은 개인정보를 저장하는 기능이다. 사용자는 회원 가입을 함으로써 For single life의 서비스를 이용할 수 있게 된다. |
| Input | 기본적으로 ID, 비밀번호, 이름, 성별, 전화번호, 주소를 입력한다. |
| Output | 모든 정보가 정확하게 입력되었을 경우 회원 가입 완료 창을 출력한다. |
| Handle | 모든 정보가 정확하게 입력되었을 경우 사용자의 정보를 User DB에 저장한다. |
| Qualification | 본인 인증을 위해 핸드폰을 이용한 인증과정을 거치도록 한다. |

**Table 4 Sign up**

**B. Login**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 로그인 |
| Explanation | 사용자가 회원가입을 통해 생성한 계정을 입력하여 For Single Life에 로그인하는 기능이다. 로그인이 성공적으로 진행되었을 경우, 사용자는 For Single Life의 서비스를 이용할 수 있게 된다. |
| Input | 사용자는 로그인 창에 ID와 비밀번호를 입력한다. |
| Output | 모든 정보가 정확하게 입력되었을 경우 사용자의 개인 페이지를 출력한다. 정보가 불일치할 경우 로그인 실패 창을 출력한다. |
| Handle | 입력 받은 정보를 User DB와 대조하여 비교한다. 일치하면 사용자의 개인 페이지로 이동하도록 하고 불일치할 경우 로그인 실패 창으로 이동하도록 한다. |
| Qualification | 5회 이상 로그인을 실패할 경우 해당 IP를 24시간동안 로그인이 불가능하도록 한다. |

**Table 5 Login**

**C. Homemade food Sales registration**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 수제 요리 판매 등록 |
| Explanation | 자신이 만들어 놓은 요리를 판매하기 위해 등록하는 기능이다. |
| Input | 사용자는 자신의 요리 이름, 설명과 사진, 들어가는 재료 및 판매가을 입력하여야 한다. |
| Output | 모든 정보가 정확하게 입력되었을 경우 등록 완료 메시지를 출력한다. 등록이 된 게시글은 사용자의 User DB에서 주소를 불러와 가까운 곳의 구매자들에게 추천 결과로 나타나게 한다. |
| Handle | 입력된 정보가 정확하게 입력되었을 경우 User DB에서 사용자의 전화번호를 연동하여 구매자들이 연락을 편리하게 할 수 있도록 한다. |
| Qualification | 들어가는 재료에 알러지 유발 위험의 재료가 있으면 표시하도록 한다. |

**Table 6 Homemade food Sales registration**

**D. Food Recommendation**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 음식 추천 |
| Explanation | 사용자가 로그인을 진행한 후에 기본적으로 바로 사용이 가능한 기능으로, 사용자가 거래 가능한 거리 안에 있는 음식들 중 추천 순위, 평가 등의 요소들을 다각적으로 고려해서 보여준다. |
| Input | 사용자가 로그인을 하면 User DB에 있는 주소지를 입력 받는다. |
| Output | 입력 받은 주소를 바탕으로 추천 알고리즘을 적용하여 나온 음식들 중 상위 5개 음식들을 출력한다. |
| Handle | 사용자의 검색 기록을 바탕으로 자주 검색하는 음식들 위주로 결과가 나오도록 한다. |
| Qualification | 만약 근거리에 판매중인 음식이 없을 경우 타 지역의 추천 결과를 볼 수 있도록 선택하는 버튼을 띄운다. |

**Table 7 Food Recommendation**

**E. Recommendation Evaluating**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 추천 평가 |
| Explanation | 사용자가 추천 받은 음식에 대한 피드백을 제공할 수 있는 기능이다. 사용자가 마음에 들지 않는 음식을 추천화면에서 삭제할 수 있다. |
| Input | 사용자가 마음에 들지 않는 음식에 대해 제거 입력을 받는다. |
| Output | 제거 입력을 받은 음식을 화면에서 삭제한다. |
| Handle | 제거된 음식은 다음 번 로그인에는 나타나지 않게 한다. |
| Qualification | 5개 추천 음식이 전부 제거되면 새로운 추천음식을 띄운다. |

**Table 8 Recommendation Evaluating**

**F. Search Food**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 음식 검색 |
| Explanation | 사용자가 추천 받은 음식 이외에 먹고 싶은 메뉴에 대해 키워드로 자신이 원하는 음식에 대해 검색을 하는 기능이다. |
| Input | 사용자가 원하는 음식의 키워드를 입력한다. |
| Output | 해당 키워드가 들어간 음식 중 사용자 주소와 가까운 순으로 정렬해서 출력한다. |
| Handle | 해당 키워드에 해당하는 음식이 등록되어 있지 않을 경우 다른 음식을 검색하도록 한다. |
| Qualification | 검색 기록은 저장이 되도록 한다. |

**Table 9 Search Food**

**G. Homemade Food Purchase**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 음식 구매 |
| Explanation | 사용자가 자신이 선택한 음식을 구입하는 기능이다. 원하는 음식이 존재할 경우 해당 음식을 등록한 판매자와 거래를 진행할 수 있다.. |
| Input | 구매하는 사용자가 원하는 음식을 선택한다. |
| Output | 해당 음식의 판매자와 연락을 할 수 있는 시스템으로 연결되어 거래하도록 한다. |
| Handle | 해당 음식의 판매자와 구매자의 연결 시스템이 이루어지도록 한다. |
| Qualification | 거래 기록은 저장이 되도록 한다. |

**Table 10 Homemade Food Purchase**

**H. Reviews and ratings**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 후기 및 평점 |
| Explanation | 사용자가 음식을 구입한 후 해당 음식 및 등록한 판매자에 대한 평가를 하는 기능이다. 이 평가를 통해 사용자들은 판매자의 신뢰성을 평가할 수 있으며 또한 추천 음식의 제시 기준이 된다. |
| Input | 구매자가 거래가 완료된 음식을 선택한다. |
| Output | 해당 음식에 대한 평점과 판매자에 대한 후기를 작성한다. |
| Handle | 여러 기록이 쌓이면 평점의 평균을 계산하도록 한다. |
| Qualification | 5점 만점의 Star Rating 방식으로 계산한다. |

**Table 11 Reviews and ratings**

**I. Recipe Confirmation**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 조리법 확인 |
| Explanation | 사용자가 다른 사용자가 등록한 음식의 레시피를 확인할 수 있는 기능이다. 등록된 레시피는 유료/무료에 따라 결제 서비스를 포함하게 된다. |
| Input | 검색한 결과를 선택했을 때 조리법 확인란을 선택할 때 나타난다. |
| Output | 해당 음식의 조리법이 적힌 페이지를 출력하도록 한다. |
| Handle | 레시피가 유료공개일 경우 결제창을 출력하도록 한다. |
| Qualification | 유료공개인 레시피는 확인란에 유료라는 표시가 나오도록 한다. |

**Table 12 Recipe Confirmation**

**J. Community**

|  |  |
| --- | --- |
| Functional Requirement | |
| Function | 커뮤니티 |
| Explanation | 사용자들 간의 소통을 제공하는 기능이다. 사용자 간의 공감대를 형성하고 서로의 정보를 나눌 수 있는 자유로운 공간이다. |
| Input | 게시글을 검색할 키워드를 입력하거나, 게시글을 작성할 경우 글의 제목과 내용을 입력하도록 하고, 댓글을 작성할 경우 댓글 내용을 입력하도록 한다. |
| Output | 키워드를 입력해서 검색할 경우 키워드가 포함된 글들을 출력하고, 게시글이나 댓글을 작성한 경우 해당 내용이 추가된 페이지를 표시하도록 한다. |
| Handle | 글이나 댓글을 작성한 경우 글은 최신 내용이 제일 위로, 댓글은 제일 아래로 정렬하도록 한다. |
| Qualification | 커뮤니티 기능을 처음 실행할 경우 게시판의 모든 글들을 등록시간의 내림차순으로 정렬한다. |

**Table 13 Community**

**6.3 Non-Functional Requirements**

비기능적 요구사항으로는 Product requirement, Organizational requirement, External requirement가 있다.

**A.Product requirement**

**A.1 Performance**

해당 시스템은 실시간 개인간 거래(Real-time base P2P trade)이므로 유연한 서 비스를 제공하여야 한다. 거래중의 과정들이 바로 확인이 가능해야 하며 내용의 누락이 없어야 하기 때문에 Performance적인 측면에서 문제가 없도록 해야 한다. 또한 사용자의 판매등록이나 등록한 음식에 대한 추천 필터 적용은 항상 최신화가 이루어져야 한다. 성능이 좋은 서버를 이용하여 이 부분에서의 품질 저하가 발생하지 않도록 해야 한다.

**A.2 Security**

해당 시스템은 개인간의 거래를 위해 사용자의 주소나 전화번호 같은 개인정보를 수집한다. 그렇기 때문에 외부의 악의적인 침입으로부터 해당 시스템을 보호하고, 고객의 개인정보가 유출되는 일이 발생하지 않도록 철저한 보안 시스템을 사용해야 한다. 고객의 개인 정보는 모두 암호화되어 침입자가 함부로 정보를 열람하지 못하도록 해야 한다. 또한 관리자를 위한 접속 경로는 관리자를 제외한 다른 사용자들의 접근을 방지하기 위해 관리자만이 알고 있도록 설계되어야 한다. 사용자의 계정 정보는 관리자를 제외하고는 사용자 본인만이 관리할 수 있도록 해야 한다.

**A.3 Efficiency**

해당 시스템은 서버와의 지속적인 통신을 전제로 한다. 실시간 상품 추천과 거래를 지원하는 시스템이기 때문에 시스템 사용 중 정보를 불러오는 대기 시간이 길어지지 않도록 해야 한다.서버와의 지속적인 통신이 불가능하다면 큰 장애요소로 이어지기 때문에 서버와의 통신 데이터는 최소화해야 하며 통신의 전환에 소요되는 시간 또한 최소화해야 한다. 또한 다량의 사용자가 동시에 통신을 요청해도 안정적으로 시스템을 운영할 수 있도록 서버를 유지해야 한다.

**A.4 Dependability**

해당 시스템은 가까운 거리의 고객 간의 거래를 가능하게 하기 위해서 실시간으로 거래가 진행되기 때문에 거래중인 상품이임의로 변경되거나 삭제되지 않도록 유지되어야 한다. 또한 판매자가 등록한 상품에 대해 수정이나 삭제를 할 경우 즉각적으로 반영되어 나타나도록 해야 한다. 따라서 사용자에게 신뢰도 높은 서비스를 제공하기 위해 안정적인 서버 구축 및 운영을 필요로 한다.

**A.5 Usability**

해당 시스템은 사용자의 사이트 이용에 도움을 줄 수 있는 기능을 제공해야 한다. 카테고리를 설정하여 사용자들이 자신이 원하는 음식을 쉽고 간편하게 검색할 수 있도록 한다. 버튼의 배치나 게시판의 구조 등은 사용자들의 직관적인 이해를 도울 수 있도록 구성되어야 한다. 따라서 UI/UX적인 측면에서의 불편함이 없도록 해야 한다. 또한 자신의 거래중인 상품이나 거래 완료된 상품에 대해 확인할 수 있는 페이지를 제공하고, 거래중의 알람 서비스를 제공한다.

**B.Organizational requirement**

**B.1 Environment**

해당 시스템은 부트스트랩(Bootstrap) 기반의 반응형 웹으로 개발되어 사용자의 환경에 최적화된 화면을 제공한다. 반응형 웹의 경우 지원하지 않는 브라우저나 버전도 있으므로, 이에 대해 사용자에게 명시를 하거나 이에 대한 대비책을 준비해야 한다. For Single Life의 시작 화면에서 이를 명시하도록 한다.

**B.2 Operation**

해당 시스템은 실시간 개인간 거래 서비스(Real-time P2P trade) 이므로 판매자와 구매자가 거래를 진행할 때 채팅과 같은 상호간의 실시간 커뮤니케이션이 원활하도록 통신을 지원해야 한다. 또한 사용자들은 커뮤니티에서 게시글이나 유/무료의 레시피를 작성하거나 검색함으로써 정보를 공유할 수 있다. 음식의 구입을 마치고 각각의 음식과 판매자에 대해 평가를 할 수 있으며또한 추천 받은 음식이 마음에 들지 않는 경우에 삭제하는 기능을 제공한다.

**B.3 Development**

개인간의 실시간 거래를 제공해야 하기 때문에 빅데이터와 머신러닝을 이용하여 일정 거리내의 사용자가 등록한 음식만이 구매자에게 보이도록 페이지를 작성하여야 한다. 또한 단말기의 종류에 영향을 덜 받는 웹 서비스를 채택하고, Django 프레임워크를 통해 웹 서비스 간의 호환성을 높이도록 한다.

**C.External requirement**

**C.1 Regulatory**

해당 시스템은 사용자의 개인정보를 입력 받는 회원가입을 해야 서비스를 이용할 수 있다. 해당 시스템은 사용자의 개인정보(거래 및 결제 정보를 포함)를 외부에 제공할 경우 사용자의 동의를 우선으로 받아야 한다. 사용자가 거래, 결제 및 추천 시스템을 이용하는데 필수적인 정보의 경우에는 회원가입 과정에서 이용약관에 해당 규정을 명시하여 동의를 받도록 한다. 그 외의 경우에는 사용자의 개인정보에 접근할 수 없도록 해야 한다.

**C.2 Safety/Security**

모든 처리가 중앙 서버를 한 번 이상 경유하기 때문에 중앙 서버가 네트워크 과부하로 인한 오작동을 하지 않도록 서버 용량 및 트래픽에 대한 예측이 빗나가지 않도록 해야 한다. 또한 해당 시스템은 고객의 개인 정보가 외부의 접근으로부터 안전하도록 암호화 등의 방법을 통해 보안성을 높여야 한다. 서버 관리자는 해당 서비스에 회원 가입하여 시스템을 이용하는 사용자들을 관리하고, 악의적인 접근을 막아야한다.

**6.3 Scenario**

**A.Join and Login Scenario**

**A.1 Initial assumption**

For Single Life 시스템을 이용하기 위해서 회원가입 절차를 거친 후 로그인을 해야 한다. 사용자는 ID, 비밀번호, 이름, 성별,주소 등의 정보를 제공 후 메일 및 아이디 중복 확인과 비밀번호 확인 후 가입을 완료한다. 사용자는 회원가입을 마치면 다음번부터는 ID와 비밀번호를 입력하여 로그인을 함으로써 서비스를 이용할 수 있다.

**A.2 Normal flow of events**

사용자는 정보 제동 동의를 한 후, ID와 비밀번호를 입력하여회원 가입을 한다. 메일 및 계정 정보의 중복 확인을 거쳐 회원가입을 완료한 다. 가입할 때 작성한 ID와 비밀번호로 로그인을하여 서비스를 이용한다.

**A.3 What can go wrong & concurrent activities**

회원 가입 시, 이미 중복된 ID를 입력할 경우 중복 경고 팝업창이 뜨며 새로운 아이디를 입력하도록 요구한다. 또한 필수로 요구하는 정보가 누락되었을 경우 회원 가입이 진행되지 않으며누락된 정보를 입력하도록 경고 팝업창이 뜬다.

로그인을 할 때 DB와 ID랑 비밀번호가 일치하지 않으면 로그인 실패 팝업창이 뜬다.

**A.4 System state on completion**

로그인에 성공하면 For Single Life의 서비스를 이용할 수 있다.

**B. Food sales registration and purchase**

**B.1 Initial assumption**

For Single Life에서 음식을 판매하고자 하는 사용자는 자신이 만든 음식을 등록하여 판매할 수 있다. 반대로 원하는 음식을 구매하고자 하는 구매자는 원하는 음식 키워드를 검색하여 가장 가까운 곳에서부터 나오는 리스트 들 중 선택하여 음식을 구매한다.

**B.2 Normal flow of events**

판매자는 음식의 가격과 간단한 레시피를 등록한다. 음식에 들어가는 재료들 중 알레르기 유발 재료가 있으면 표기하도록 한다. 구매자는 원하는 음식을 검색하여 해당 가격을 지불한 다음결제가 완료되면 상호간의 거래가 이루어진다. 거래는 구매자에게 판매자의 안심번호 연락처가 제공되어 커뮤니케이션이 이루어진 다음 이루어진다.

**B.3 What can go wrong & concurrent activities**

결제가 제대로 이루어지지 않거나 구매자가 도중에 취소를 할 경우 일정 비율의 수수료를 지불하여 판매자에게 지급되도록 한다.

**B.4 System state on completion**

거래가 완료되면 판매자에게는 판매 금액이, 구매자에게는 후기를 남길 수 있는 창이 생긴다.

**C. Review and Rating**

**C.1 Initial assumption**

구매를 완료한 음식을 대상으로 음식에 대한 후기와 판매자에 대한 평가를 남길 수 있다. 평가 결과들은 누적되어 평균 점수가 공개되며 차후 다른 구매 예정자들이 평가할 수 있는 지표가 된다.

**C.2 Normal flow of events**

거래가 완료되면 구매한 음식을 대상으로 평가를 할 수 있는 칸이 생성된다. 구매한 음식과 판매자는 각각 5점 만점의 Star Rating 방식으로 평가를 할 수 있으며 점수 이외에도 의견을 개진할 수 있는 항목이 있다. Star Rating Point는 평균점수를 계산하여 차후에 검색하는 구매자들에게 보이게 된다.

**C.3 What can go wrong & concurrent activities**

후기에 이유 없는 비방이나 비하, 욕설 등의 부적절한 내용이들어가 있을 경우 운영자의 판단으로 통보없이 삭제될 수 있다.

**C.4 System state on completion**

후기를 작성하고 나면 그 내용이 저장되어 다른 이용자들도 후기를 볼 수 있게 된다.

**D.Community**

**D.1 Initial assumption**

For Single Life의 커뮤니티에서 서로의 의견을 자유롭게 교환할 수 있다. 유/무료의 레시피를 등록하거나, 음식에 대한 전반적인 의견을 교환하는 정보 교류 역할을 한다. 게시글을 작성하거나 댓글을 작성함으로써 의견을 개진할 수 있다.

**D.2 Normal flow of events**

레시피를 공유하고자 하는 이용자는 유/무료로 자신의 레시피를 공개할 수 있다. 유료 레시피를 열람하고자 하는 이용자는 해당 가격을 지불하여 레시피를 열람할 수 있다. 게시글이나 댓글은 등록되면 게시글은 최근 올라온 글부터, 댓글은 먼저 등록된 댓글부터 정렬되어 출력된다.

**D.3 What can go wrong & concurrent activities**

레시피에 문제가 있어서 신고가 누적되거나, 부적절한 내용이담긴 게시글이나 댓글을 작성한 경우에는 해당 사용자에게 통보없이 운영진의 판단 하에 삭제가 이루어 질 수 있다.

**D.4 System state on completion**

해당 게시글, 혹은 레시피가 작성되고 나면 모든 이용자들이 유/무료로 해당 내용을 볼 수 있게 된다.

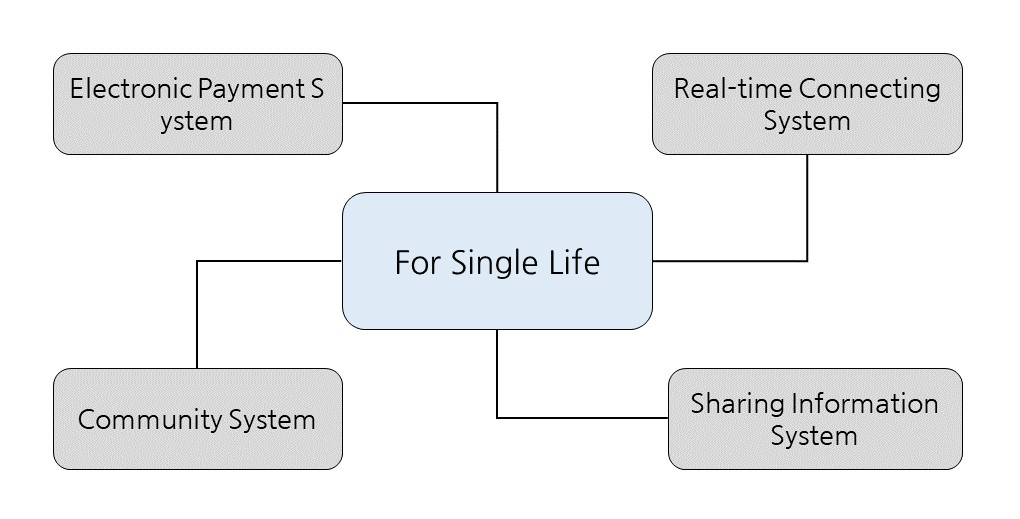
**7. System Models**

**7.1. Objective**

System Models 에서는 특정 기능을 사용할 때 시스템 컴포넌트, 시스템 간의 상호작용에 대해 기술한다. 자세한 설명을 위해 UML을 사용한 context diagram, data-driven model, event-driven model, interaction model 등으로 나타내어 시스템을 표현하였다.

**7.2. Context models**

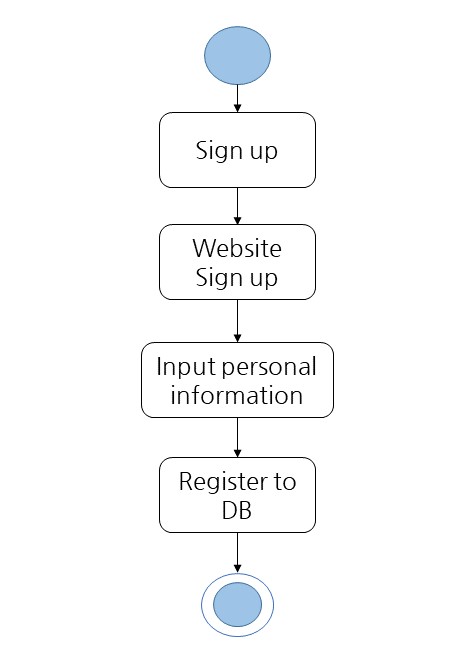
**A. Context Model**



**Diagram 8 Context model**

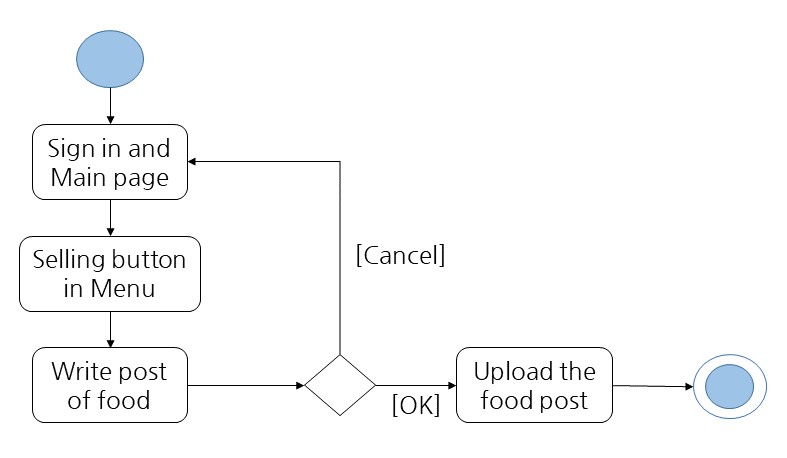
**B. Process Diagram**

**B-1. Sign up process**



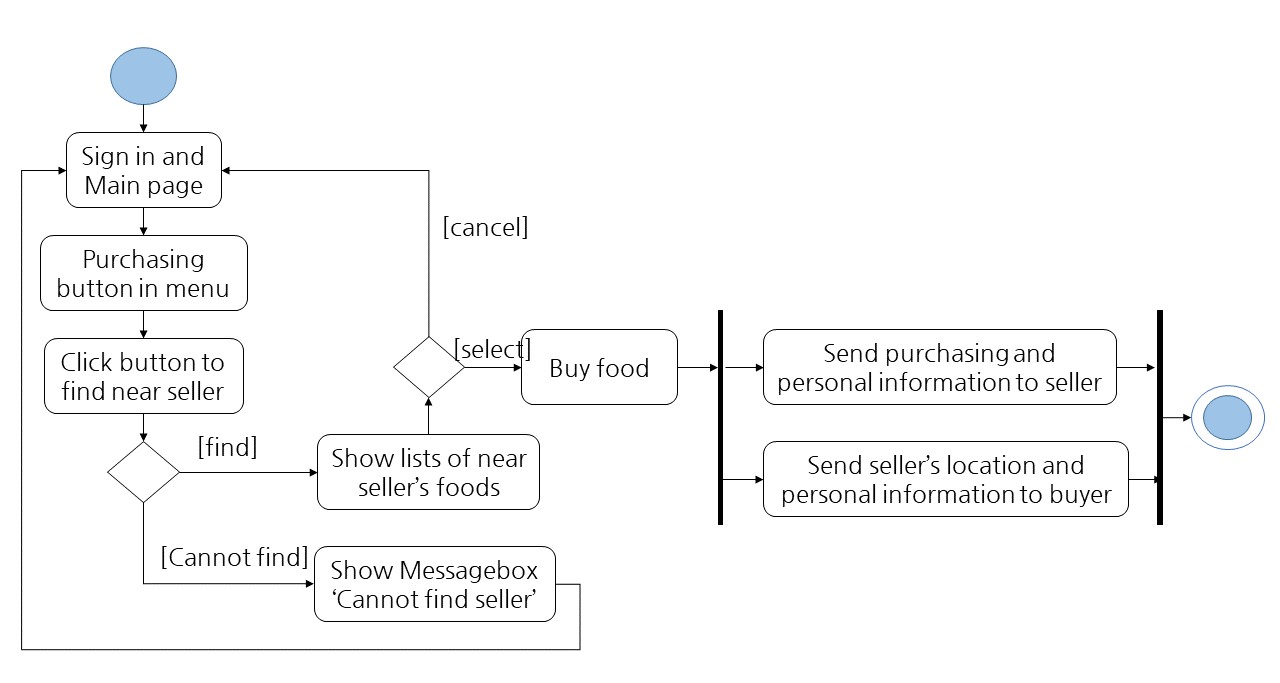
**Diagram 9 Sign up process**

**B-2. Selling process**



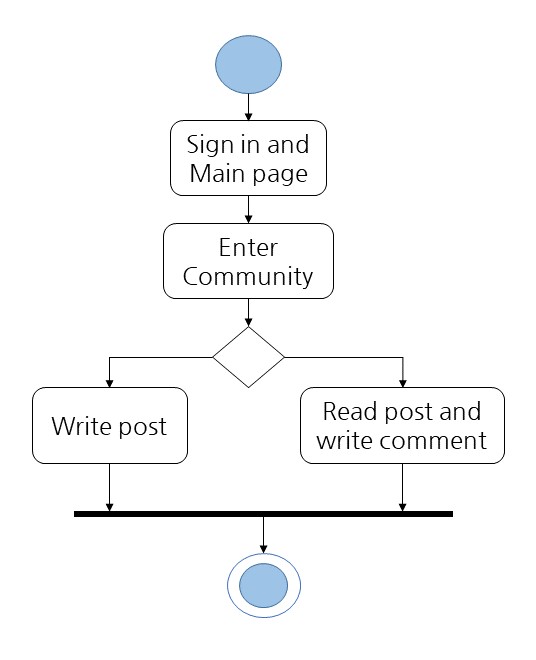
**Diagram 10 Selling process**

**B-3. Purchasing process**



**Diagram 11 Purchasing process**

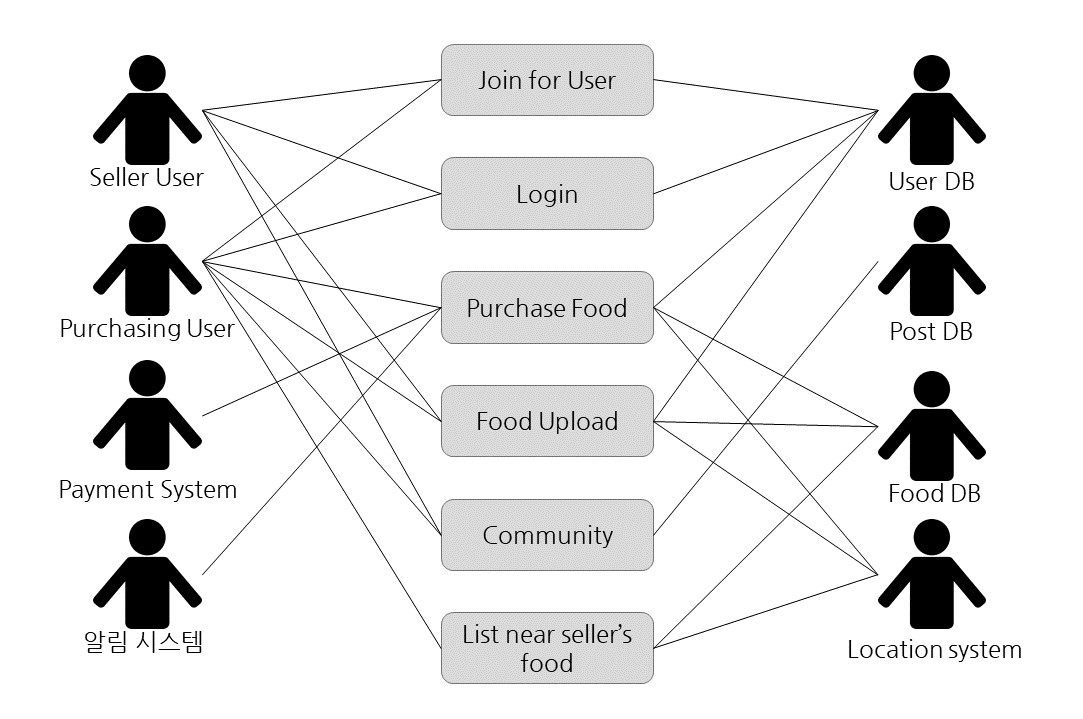
**B-4. Community process**



**Diagram 12 Community process**

**7.3. Interaction models**

**A. Use case Models**



**Diagram 13 Use Case Diagram**

**B. Tubular description for each use case**

**B-1. Join for User**

|  |  |
| --- | --- |
| USE CASE | Join for User |
| ACTOR | Seller User, Purchasing User, User DB |
| DESCRIPTION | - 회원가입을 위해서 User의 개인 정보를 요청하고 입력받은 후에 이 정보들을 User 데이터베이스에 저장하는 기능  - User들은 ‘For Single Life’서비스를 이용하기 위해서 회원가입을 진행할 때 반드시 회원정보에 휴대폰 번호를 기입 |
| DATA | User의 ID/PW정보와 개인정보 |
| STIMULUS | 서비스 실행 시 로그인 화면에서 회원가입 요청 후 회원가입 실행 시 필요한 정보 입력 요청 |
| RESPONSE | - 회원가입에 필요한 정보들을 양식으로 화면에 출력  - User가 입력한 양식이 올바른 정보일 경우에 User Database에 업데이트 한 후 완료 메시지 출력 |

**Table 7-1. Tubular Table : Join for User**

**B-2. Login**

|  |  |
| --- | --- |
| USE CASE | Login |
| ACTOR | Seller User, Purchasing User, User DB |
| DESCRIPTION | - 회원가입을 한 User가 서비스를 사용하기 전에 로그인 페이지에 입력한 ID/PW와 데이터베이스에 입력된 정보를 비교  - 서비스 이용 가능 여부를 판단 |
| DATA | User의 ID/PW 정보 |
| STIMULUS | 서비스 실행 시 로그인 화면에서 로그인 요청 |
| RESPONSE | - 로그인 페이지에 로그인에 필요한 ID/PW 입력 양식 출력  - User가 입력한 정보가 User Database에 저장된 정보와 비교  - 일치할 경우 로그인 성공 메시지 출력, 일치하지 않을 경우 실패 메시지 출력 |

**Table 7-2. Tubular Table : Login**

**B-3. Purchase Food**

|  |  |
| --- | --- |
| USE CASE | Purchase Food |
| ACTOR | Purchasing User, User DB, Food DB, Payment System, 알림 시스템, Location System |
| DESCRIPTION | - 구매 User가 판매 User가 판매하는 음식을 구매하는 서비스  - 구매한 후 해당 음식의 판매 User와 구매 User에게 각각 서로의 개인정보 및 음식 구매정보 전송 |
| DATA | User의 결제 수단 |
| STIMULUS | 원하는 음식의 상세 페이지에서 구매 버튼을 클릭하면 결제 요청을 전송 |
| RESPONSE | - 판매 User 위치 정보와 결제 금액을 결제 진행 페이지에서 출력 후 결제 시스템에 결제 요청  - 결제가 완료되면 결제 성공 메시지 출력 후, Food DB를 업데이트하여 해당 음식의 수량을 결제 완료 수량만큼 차감  - 결제와 동시에 판매 User와 구매 User의 휴대폰으로 서로의 개인정보 및 위치정보 그리고 음식 결제정보 알림 |

**Table 7-3. Tubular Table : Purchase Food**

**B-4. Food Upload**

|  |  |
| --- | --- |
| USE CASE | Food Upload |
| ACTOR | Purchasing User, Selling User, Food DB |
| DESCRIPTION | - 판매 User가 음식을 판매하기 위해 진행하는 서비스  - 판매 User는 음식의 가격과 수량, 음식 사진 등등 음식의 상세한 정보들 업로드 가능 |
| DATA | User의 음식 가격, 수량, 미디어 파일 |
| STIMULUS | 판매 User가 음식에 대한 글을 업로드 요청 |
| RESPONSE | - 판매 User가 음식 상세정보를 올리면 Food DB에 판매 User가 올린 음식 이름, 가격, 수량, 위치를 저장 |

**Table 7-4. Tubular Table : Food Upload**

**B-5. Community**

|  |  |
| --- | --- |
| USE CASE | Community |
| ACTOR | Purchasing User, Selling User, Post DB |
| DESCRIPTION | - 가입한 모든 User들이 자유롭게 음식에 대한 레시피나 이야기를 작성하거나 읽을 수 있는 서비스  - 사진이나 동영상 같은 미디어를 활용하여 글 작성 가능 |
| DATA | User의 글, 미디어파일 등등 |
| STIMULUS | User가 Post 작성 후에 업로드 버튼 클릭 시 Post DB에 요청 |
| RESPONSE | - User의 글이나 미디어 파일 등등을 Post DB에 새로 저장  - 성공적으로 저장되었을 경우 성공 메시지 출력, 저장에 실패하였을 경우 실패 메시지 출력 |

**Table 7-5. Tubular Table : Community**

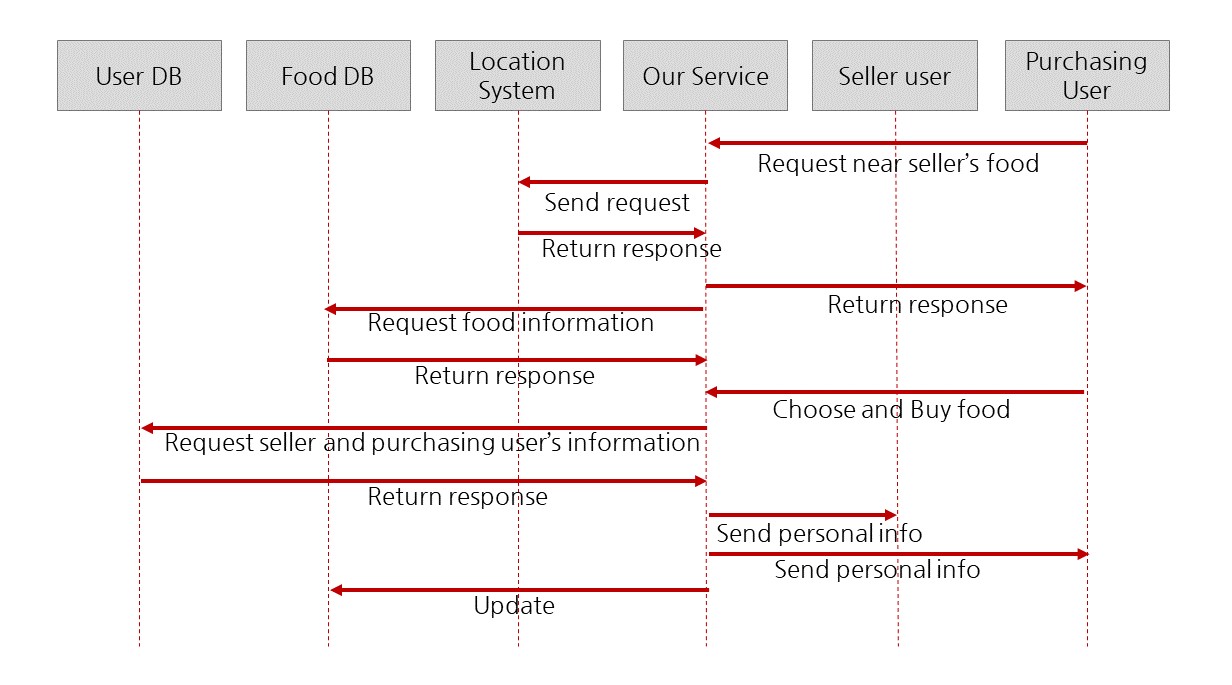
**B-6. List Near Seller’s Food**

|  |  |
| --- | --- |
| USE CASE | List Near Seller’s Food |
| ACTOR | Purchasing User, Food DB, Location System |
| DESCRIPTION | - 구매 User가 음식 구입하기 버튼을 클릭하였을 때 구매 User의 위치 정보를 이용하여 가까운 곳에서 음식을 판매하고 있는 User들을 검색해 주는 서비스  - 가까운 구매자들의 음식 정보들을 List 형식으로 나열하여 구매 User에게 제공  - List에 있는 어떤 음식을 클릭할 경우에 음식에 대한 상세정보 조회 가능 |
| DATA | 구매 User의 위치 정보 |
| STIMULUS | 구매 User가 음식 구매 버튼을 누르면 구매 User의 위치정보와 검색 요청 전송 |
| RESPONSE | - 구매 User의 위치정보와 현재 판매하고 있는 판매 User가 음식을 업로드할 때 입력했던 위치를 계산하여 가까이에 있는 판매 User들의 음식을 Food DB에서 List로 구매 가능 음식 결과 출력  - List의 음식 중 하나를 클릭할 경우 해당 음식의 상세 정보 페이지 출력 및 결제 버튼 출력  - 가까이에 있는 판매 User가 없을 경우 구매 가능 음식 검색 실패 메시지 출력 |

**Table 7-6. Tubular Table : List Near Seller’s Food**

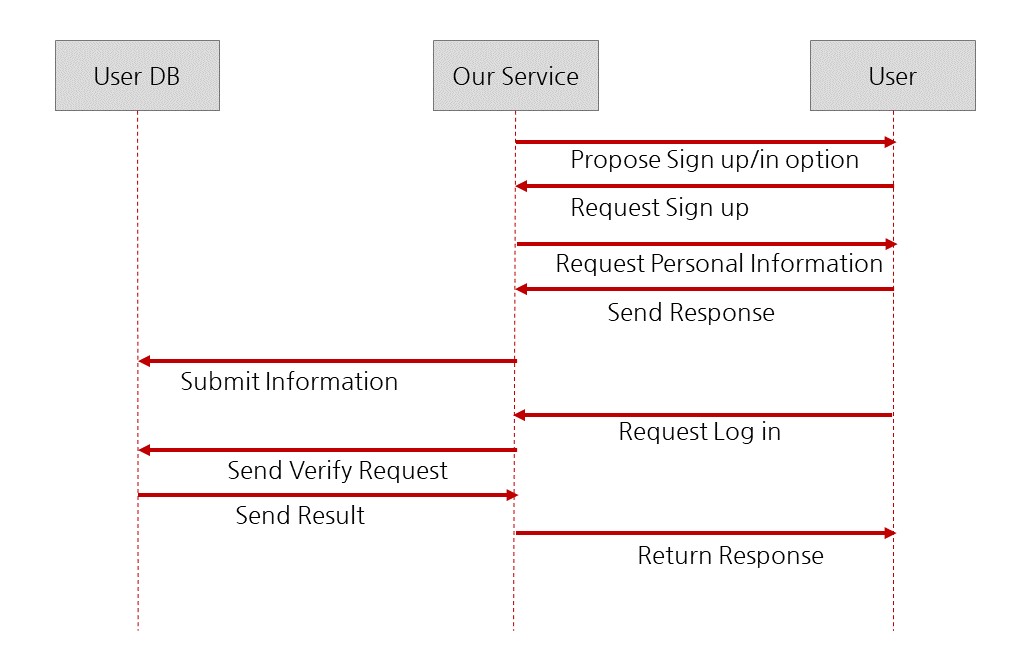
**C. Sequence Diagram**

**C-1. Sequence Diagram of Purchasing Food**



**Diagram 14 Sequence Diagram of Purchasing Food**

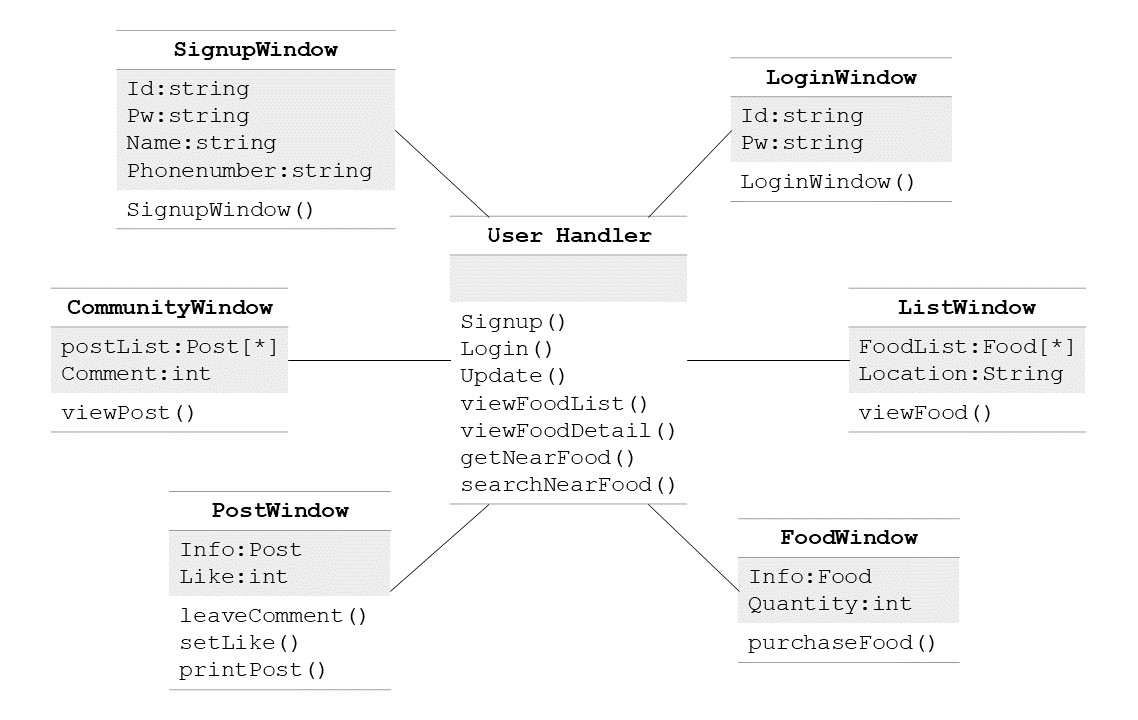
**C-2. Sequence Diagram of Log-in and Sign-up**



**Diagram 15 Sequence Diagram of Log-in and Sign-up**

7.4. Structural models

A. User System Class Diagram



**Diagram 16 User System Class Diagram**

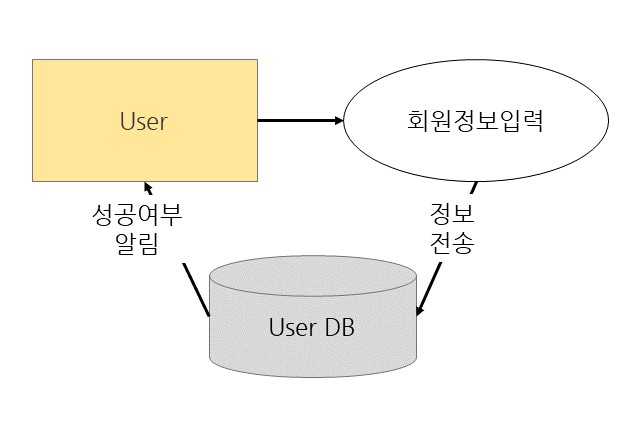
7.5. Behavioral Models

A. Data-Driven Modelling

본 시스템(For Single Life)의 주요 기능인 음식 구매 시스템, 가까운 판매자의 음식 검색 시스템, 알림 시스템에 중점을 두어 Data-Driven Model을 작성하였다.

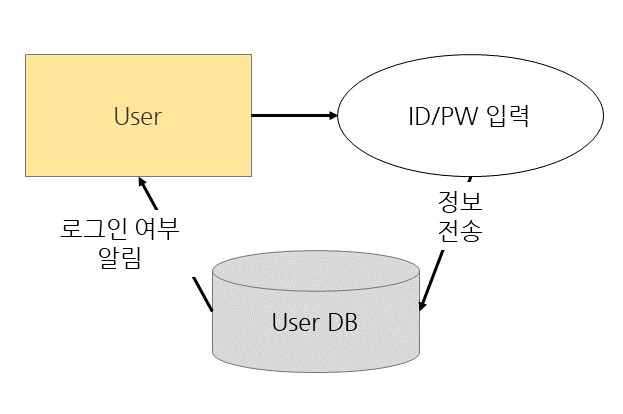
**A-1. User**

**- Sign-up**



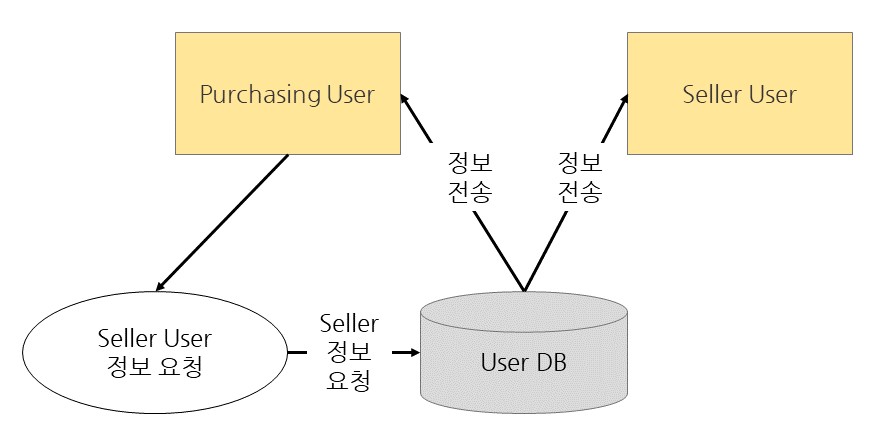
**Diagram 17 Data Flow Diagram of User : Sign-up**

**- Log-in**



**Diagram 18 Data Flow Diagram of User : Log-in**

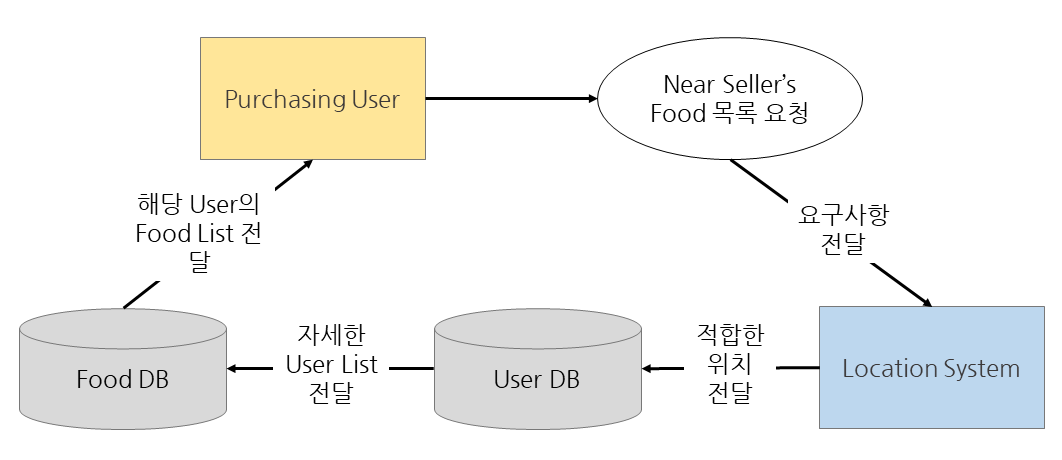
**- 개인정보 공유**



**Diagram 19 Data Flow Diagram of User : 개인정보 공유**

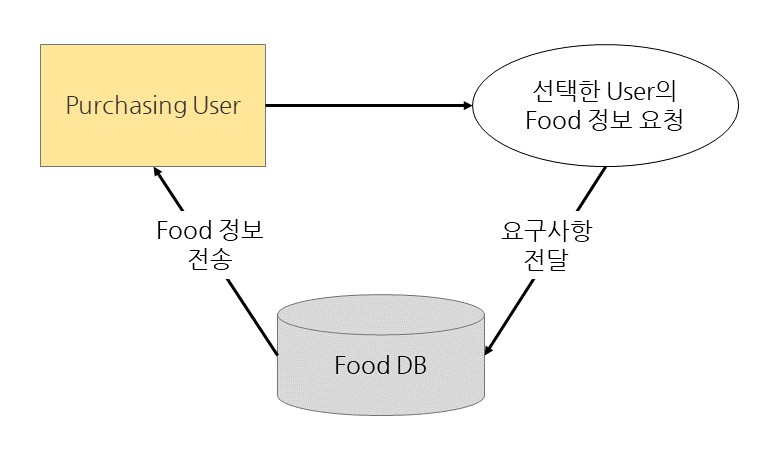
**A-2. Purchasing User**

**- 구매 가능 음식 목록 확인**



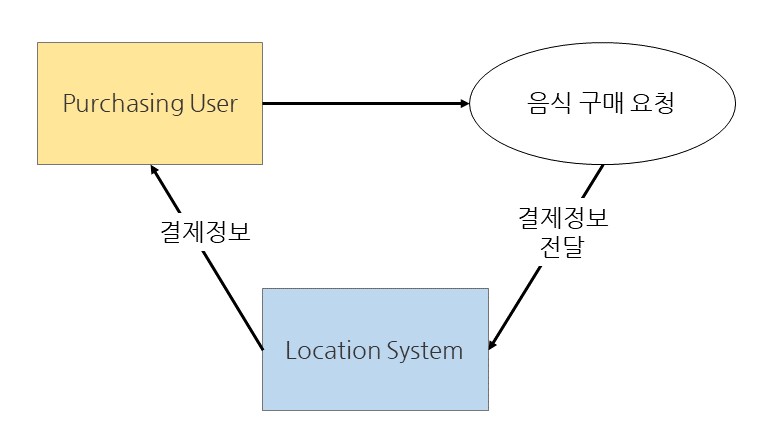
**Diagram 20 Data Flow Diagram of Purchasing User : 구매 가능 음식 목록 확인**

**- 음식 상세정보 조회**



**Diagram 21 Data Flow Diagram of Purchasing User : 음식 상세정보 조회**

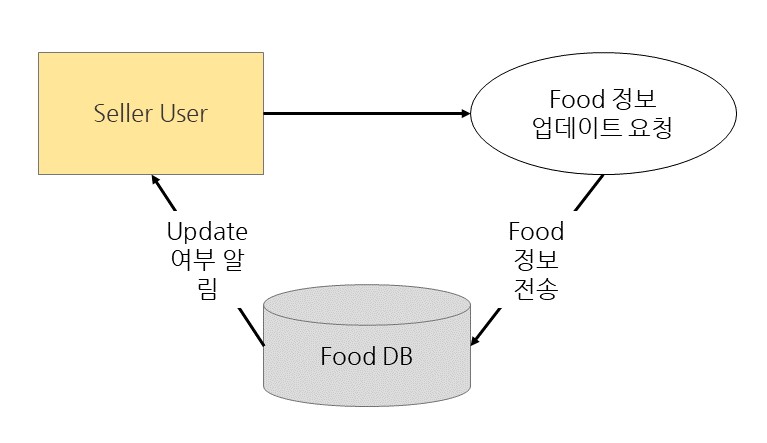
**- 구매 요청**



**Diagram 7-14. Data Flow Diagram of Purchasing User : 구매 요청**

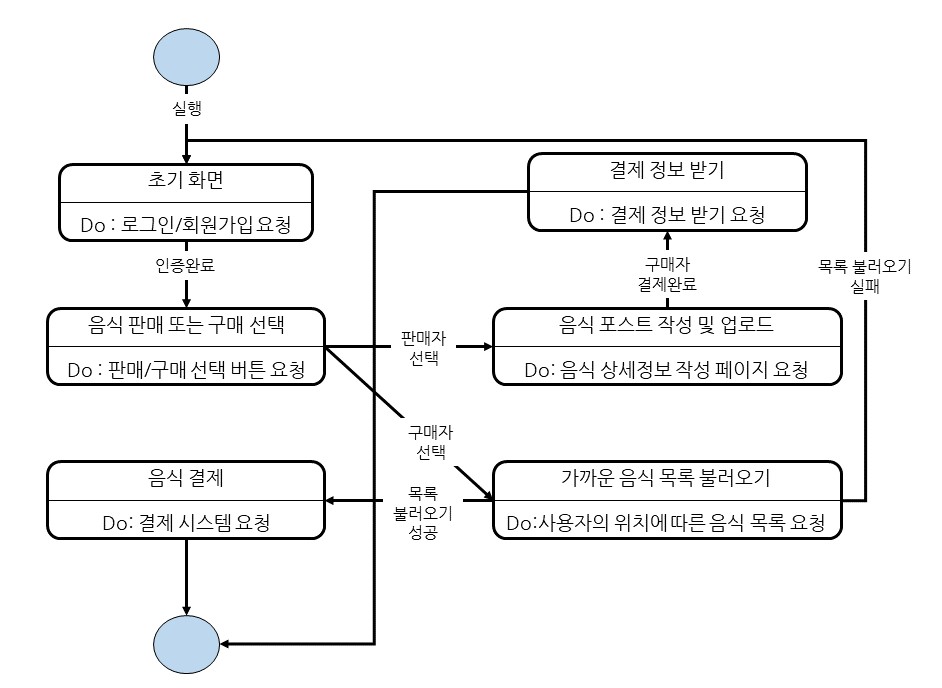
**A-3. Seller User**

**- 음식 정보 업로드**



**Diagram 22 Data Flow Diagram of Seller User : 음식 정보 업로드**

**B. Event-Driven Modelling**



**Diagram 23 ‘For Single Life’ State Diagram**

**8.System Evolution**

**8.1 Objective**

This part of system evolution describes in detail the evolution of the system in terms of software, the evolution according to customer needs, and the evolution according to the environment. It is hoped that the content described in this chapter will give more help to other teams who want to develop similar software.

**8.2 Limitation and Assumption**

There are many similar takeaway software nowadays, especially in China, where similar software has achieved great success, and even many people regard takeaway as their main occupation. Therefore, we hope that the software we independently developed can better adapt to the market and provide high-quality services for both businesses and customers.

Due to the limitation of software, the delivery and payment of takeout are only settled separately by merchants and customer service. There is also the time required to deliver meals, these factors are uncontrollable, decided by the business.

**8.3 Evolutions of Software**

**A.Video Streaming**

Video streaming refers to the transmission of video data. Through video shooting and video uploading, buyers can get a clearer understanding of the goods, and there will not be frequent inconsistencies between photos and objects. This canensure the authenticity of the goods, customers can also more quickly understand the information of the goods, including the weight, style, will not appear after buying the situation do not want to eat. This will help businesses to promote and safeguard the interests of customers.

1. **Big Data Statistics**

Statistics of big data can help businessmen understand the market, but also help customers choose. Choose some popular products and special products, including some new products, as far as possible to provide customers with more choices.

**8.4 Evolutions of User Requirement**

**A. Business Needs**

Businessmen need more feedback from consumers, including the taste of food, the amount of food, the integrity of packaging and so on. This information can help businesses better understand the needs of customers, so that users'needs are understood.

**B. Buyer's Demand**

Software needs to collect information about goods that buyers often buy and pay attention to, to provide recommendations for similar goods for buyers, so that buyers have more choices. The system also needs to be able to push some new products for buyers and sell them hot so that buyers can choose easily.

**8.5 Evolutions of Environment**

Now the software can display the goods and push them. In order to understand the goods more intuitively and help customers choose the goods, it is necessary to offer after-sales evaluation service as far as possible. Buyers can comment directly under the goods, others can also see the evaluation, can be more intuitive, fast understanding of goods information.

**9.Appendices**

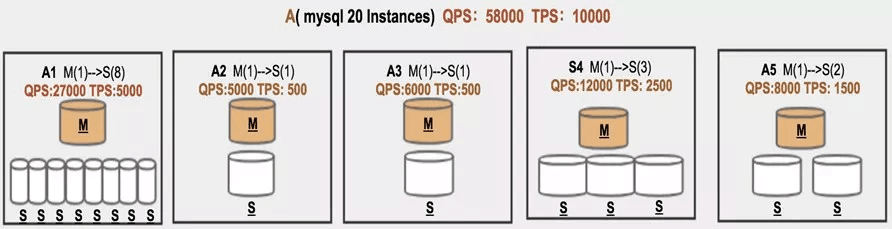
**8.1 Objective**

This chapter introduces more specific information about the system of For Single Life. System settings, databases, and development processes used in this system.

**9.2 Database requirements**

The system needs to design an analytical database to obtain user data, analyze data and feedback more information to users.

With the increase of users, there is a growing demand for business, and it is faced with the need to split the hot business, need to separate the hot business, a set of databases into multiple sets, vertical business splitting.



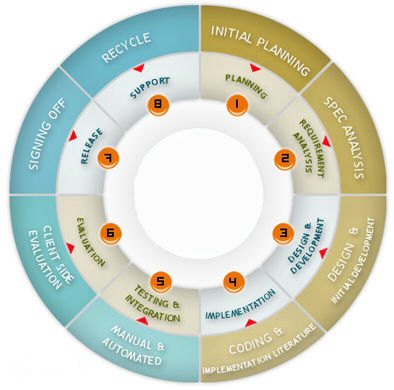
**Figure 8 Split Vertically**

**9.3 User-System Requirements**

For the user system requirements, businesses should be able to receive orders from consumers during business hours, businesses can launch their own recommended products, new products and so on. The interface is simple, the classification is clear and easy to operate.

For consumers, it is also necessary to keep the real-time update of business information synchronized, the range of meals is accurate, the address of meals is error-free, and the telephone input is correct. The goal of this system is to accurately collect customer needs, match to the corresponding businesses, and display the most suitable businesses on the consumer system according to the location of consumers.

**9.4 Development process**



**Figure 9 Ailge**

This software system is quite complex, in addition to the most basic content, including video streaming, database, location. There are many problems in the development process, and it is difficult to give a complete definition of the program before development.

Therefore, this project applies agile methodology for development. After expanding the development process, we should first develop the most priority part. After the development of these parts is completed, the new parts are redeveloped, improved and developed through feedback to the development. If the new part is developed, the feedback will increase gradually. In the development process, if the overall feedback work is needed, it is reconstructed according to the inspection.

**10.Index**

**10.1** **Figure Index**

Figure 1 Single family status and forecasts----------------------------------------------12

Figure 2 The problem of malnutrition among those living alone in news-------------14

Figure 3 The problem of malnutrition among those living alone in internet searching-----------------------------------------------------------------------------------------------15

Figure 4 Log-in & Sign up’ System-------------------------------------------------------32

Figure 5 real time recommendation system----------------------------------------------33

Figure 6 purchase system-----------------------------------------------------------------34

Figure 7 Community system--------------------------------------------------------------35

Diagram 8 Context model---------------------------------------------------------------42

Diagram 9 Sign up process---------------------------------------------------------43

Diagram 10 Selling process----------------------------------------------------------43

Diagram 11 Purchasing process-----------------------------------------------------44

Diagram 12 Community process----------------------------------------------------45

Diagram 13 Use Case Diagram-----------------------------------------------------46

Diagram 14 Sequence Diagram of Purchasing Food------------------------------50

Diagram 15 Sequence Diagram of Log-in and Sign-up--------------------------51

Diagram 16 User System Class Diagram-------------------------------------------52

Diagram 17 Data Flow Diagram of User : Sign-up-------------------------------53

Diagram 18 Data Flow Diagram of User : Log-in---------------------------------53

Diagram 19 개인정보 공유-----------------------------------------------------------54

Diagram 20 구매 가능 음식 목록 확인---------------------------------------------54

Diagram 21 음식 상세정보 조회----------------------------------------------------55

Diagram 22 음식 정보 업로드-------------------------------------------------------56

Diagram 23 ‘For Single Life’ State Diagram---------------------------------------57

Figure 8 Split Vertically--------------------------------------------------------------------60

Figure 9 Ailge------------------------------------------------------------------------------61

**10.2 Table Index**

Table 1 User Terms------------------------------------------------------------------------20

Table 2 Service Terms---------------------------------------------------------------------21

Table 3 Develop computation-related terms---------------------------------------------21

Table 4 Sign up---------------------------------------------------------------------------36

Table 5 Login------------------------------------------------------------------------------37

Table 6 Homemade food Sales registration--------------------------------------------38

Table 7 Food Recommendation----------------------------------------------------------39

Table 8 Recommendation Evaluating----------------------------------------------------40

Table 9 Search Food----------------------------------------------------------------------41

Table 10 Homemade Food Purchase----------------------------------------------------41

Table 11 Reviews and ratings------------------------------------------------------------42

Table 12 Recipe Confirmation------------------------------------------------------------43

Table 13 Community----------------------------------------------------------------------44

Table 7-1. Tubular Table : Join for User--------------------------------------------46

Table 7-2. Tubular Table : Login-----------------------------------------------------47

Table 7-3. Tubular Table : Purchase Food------------------------------------------47

Table 7-4. Tubular Table : Food Upload--------------------------------------------48

Table 7-5. Tubular Table : Community----------------------------------------------49

Table 7-6. Tubular Table : List Near Seller’s Food---------------------------------49

**11.Reference**

Software Evolution Process, <https://www.ukessays.com/essays/information-systems/software-evolution-process.php>

나홀로 대학생 급증… 영양상태 적신호 <http://www.hynews.ac.kr/news/articleView.html?idxno=527>

1인가구 보고서 <https://choonsik.blogspot.com/2017/02/2017-1.html>