

# Design and Implementation of a Wine Recommendation Mobile Application

Si-Myung Park<sup>†</sup> · So-Young Yoon<sup>†</sup> · Eun-Be Seo<sup>†</sup> · Jong-Seo Son<sup>††</sup>  
So-Hyun Park<sup>†††</sup> · Young-Ho Park<sup>††††</sup>

## ABSTRACT

Recently, there is an increasing number of startups in wine business because of the rapid growth of wine consumption. The entrepreneurs need support to prepare infrastructure of wine business. Entrepreneurship is a course of preparing funding, personnel, and technology for preliminary entrepreneurs. In this paper, we implement a wine recommendation application for preliminary entrepreneurs of preparing the wine business as part of the technical support project. The proposed application is to collect accurate data using barcode recognition technology. Finally, proposed application aims the building of the wine knowledge-base through the data analysis by applying efficient algorithms.

**Keywords :** Wine, Entrepreneurship, Mobile Application, Android

## 와인추천 모바일 어플리케이션 설계 및 구현

박 시 명<sup>†</sup> · 윤 소 영<sup>†</sup> · 서 은 비<sup>†</sup> · 손 종 서<sup>††</sup> · 박 소 현<sup>†††</sup> · 박 영 호<sup>††††</sup>

## 요 약

최근 국내 와인 소비문화가 빠르게 성장하면서 와인 사업 창업자들이 증가하는 추세이다. 이에 따라 와인 사업을 준비하는 예비 창업자들을 지원하기 서비스가 필요해지고 있다. 앙트러프러너십은 예비 창업자들이 자금, 인력, 기술 등을 준비하는 단계이다. 본 논문에서는 와인 사업을 준비하는 예비 창업자들을 위한 기술적인 지원사업의 일환으로 와인 추천 어플리케이션을 설계 및 구현한다. 본 논문에서는 바코드 인식 기술을 이용하여 보다 정확한 와인정보를 수집하고 수집된 와인데이터에 와인을 마시는 최적상황을 추출하는 알고리즘을 적용하여 최종적으로 사용자에게 적합한 와인을 추천한다.

**키워드 :** 와인, 앙트러프러너십, 모바일 어플리케이션, 안드로이드

## 1. 서 론

최근 현대인들의 삶 속에서 와인을 접할 수 있는 환경이 많아지면서 와인에 대한 선호도가 다양해지고 수요가 증가하고 있다. 와인에 대한 전문 지식을 갖춘 매니아 층부터 단순히 와인을 좋아하는 초보자까지 와인을 즐기고 맛보는 타겟 층은 다양해졌을 뿐만 아니라, 맛집 관련 프로그램의 열풍으로 와인은 음식과 함께 즐길 수 있는 최고의 주류로

인식되며 와인 소비는 지난 10년 간 약 38% 증가하며 다양한 목적으로 구매되고 있다[1]. 이처럼 국내 와인 소비문화가 빠르게 성장하면서[2], 와인 관련 사업의 창업자도 증가하고 있는 추세이며 와인 사업을 준비하는 예비 창업자들을 지원하기 위한 와인을 구매 할 때의 선택지표를 제공하는 서비스가 필요해지고 있다[8].

기존에 와인 사업을 위한 응용 사례로는 와인의 라벨을 인식하여 기본적인 와인 정보를 제공하는 '네이버 라벨 인식 기능'과 와인 관리 어플리케이션인 'Vivino'가 있다. 하지만 '네이버 라벨 인식 기능'은 와인 라벨 인식 기능의 인식이 정확하지 않으며[3], 'Vivino'는 전문적인 지식베이스의 부재로 정확한 시장 분석이 되지 않아 예비 창업자에게 필요한 정보를 제공하기에는 부족하다는 단점이 있다.

본 논문에서는 위의 어플리케이션의 한계점을 해결하고자 다음과 같은 어플리케이션을 제안한다. 본 논문에서는 바코

※ 이 논문은 2014년도 중소기업청의 대학 기업가센터 사업비로 숙명여자대학교 앙트러프러너십센터의 지원을 받아 연구되었음.

† 비 회 원 : 숙명여자대학교 멀티미디어과학과 학사과정

†† 비 회 원 : 숙명여자대학교 글로벌서비스학부 앙트러프러너십전공 조교수

††† 준 회 원 : 숙명여자대학교 멀티미디어과학과 석사과정

†††† 종 신 회 원 : 숙명여자대학교 멀티미디어과학과 교수

Manuscript Received : November 25, 2015

First Revision : January 8, 2016

Accepted : January 12, 2016

\* Corresponding Author : Young-Ho Park(yhpark@sm.ac.kr)

드 인식 기술을 이용하여 보다 정확한 와인정보를 수집하고 수집된 와인데이터에 와인을 마시는 최적상황을 추출하는 알고리즘을 적용하여 최종적으로 사용자에게 적합한 와인을 추천한다. 제공하는 콘텐츠 어플리케이션은 10만 중에 이르는 와인의 데이터를 자체적으로 수집하여 통계 및 분석할 수 있고 데이터를 기반으로 선호도를 추출한 Top-k의 와인 목록을 제공 할 수 있다는 장점이 있다. 또한, 바코드 인식이라는 간편 기능을 통해 해당 와인의 정보를 오프라인에서 모바일로 빠르게 제공할 수 있다.

앙트러프러너십 과정은 창업예정자들이 자금, 인력, 창업 기술 등을 준비하는 단계를 뜻한다. 본 논문에서는 앙트러프러너십에 끼치는 영향은 다음과 같다. 첫째, 앙트러프러너십 과정의 예비 창업자들에게 기반 기술 지원 측면에서 와인 사업을 위한 데이터를 제공하는 모바일 콘텐츠 어플리케이션을 구현한다. 둘째, 창업자 및 고객들에게 맞춤형 와인을 추천 해주는 기능을 통해 예비 창업자들에게 고객의 니즈를 파악할 수 있는 지식베이스를 제공하고자 한다.

본 논문의 공헌은 다음과 같다.

- (1) 앙트러프러너십 과정에 있는 와인 사업을 준비하는 예비 창업자들에게 기술적인 기반과 지식베이스를 제공한다.
- (2) 고객 니즈에 부합하는 와인 정보를 제공하여 맞춤형 서비스를 제공한다.
- (3) 바코드 인식을 통한 효율성 높은 검색 기능을 통해, 언제 어디서든 이용할 수 있는 간편한 접근 방법을 제공하여 고객 확보 및 유지를 할 수 있다.

본 논문에서의 구성은 다음과 같다. 2장에서는 관련 응용 사례를 소개하며, 3장에서는 제안하는 모바일 콘텐츠 어플리케이션의 기능과 구조도를 설명한다. 4장에서는 설계 및 구현을 위한 알고리즘을 설명한다. 5장에서는 구현 결과를 설명하며 마지막 6장에서는 본 논문의 결론을 맺는다.

## 2. 관련 연구

본 장에서는 와인에 대한 정보를 제공하고 있는 기존의 응용 사례들을 소개 및 분석한다.

### 2.1 네이버 와인 라벨 검색

본 절에서는 네이버의 와인 라벨 검색 서비스에 대해 소개한다. 와인 라벨 검색[6]은 읽기 힘든 와인라벨에 대해 좀 더 쉽게 검색할 수 있도록 도와주는 서비스이다. Fig. 1은 네이버 와인 라벨 검색 실행 화면이다. 사용자가 직접 와인의 라벨을 정면으로 사진 촬영을 하면 Visual search 기술 [7]을 이용하여 사용자가 검색한 라벨에 대한 와인의 정보를 검색하여 사용자에게 보여준다. 네이버와인에 관련된 정보를 간단하게 보여준다는 장점이 있지만 와인을 구매하는데 상대적으로 덜 결정적인 도수, 생산지 등에 대한 형식적 정보만을 보여준다는 단점이 존재한다.

### 2.2 Vivino 어플리케이션

이 절에서는 Vivino을 소개한다. 2.1절의 네이버 어플리케이션과 마찬가지로 와인라벨을 인식하여 와인을 검색하는 ‘와인 라벨 검색’이라는 기능을 이용해서 와인 검색을 제공한다[4]. Fig. 2는 Vivino 어플리케이션에서 한 와인의 정보 화면이며, 하단의 버튼으로 라벨 인식을 통한 검색을 제공한다. 와인과 관련된 정보를 간단하게 보여주며, 와인을 구매하는 상황을 고려하여 가까운 지역에서 파는 와인에 대한 정보를 제공한다. 하지만 이와 관련해서 정확하지 않은 경우가 있어 문제가 되고 있다[5]. 또한, 사용자로부터 와인에 대한 정보를 입력받는다. 사용자는 내가 마신 와인에 대한 와인가격, 구매한 곳, 평점, 시음 노트 등의 정보를 입력할 수 있다. 이렇게 입력된 와인에 대한 정보들은 친구를 맺은 사람들과 공유할 수 있으며, 사용자가 마신 와인에 대한 평점을 기반으로 와인을 사용자에게 추천한다.

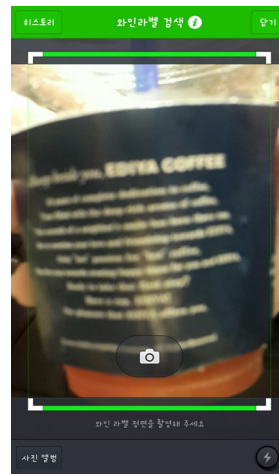


Fig. 1. Naver Wine Label Search

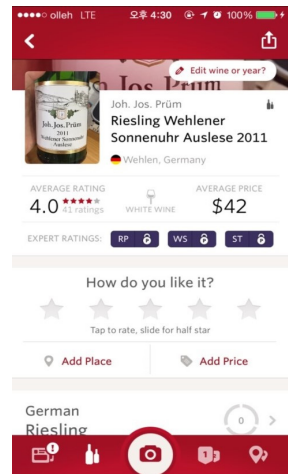


Fig. 2. Vinino Screen

## 3. 시스템 구성도

본 장에서는 시스템 구성도 및 데이터 베이스 설계에 대해 소개한다. 3.1절에서는 시스템 구성도를 소개하고 3.2절에서는 데이터베이스를 소개한다.

### 3.1 시스템 구성도

본 절에서는 본 논문에서 구현한 응용 설계에 대한 시스템 구성도를 소개한다.

Fig. 3은 클라이언트와 서버 간에 어떠한 정보를 주고받는지에 대해 간략하게 보여주는 시스템 구성도이다. 클라이언트에서 사용자의 아이디와 비밀번호, 이름 등과 같은 기본적인 사용자 정보를 웹 서버를 통해 데이터베이스에 저장한다. 새로운 와인에 대한 사진을 촬영할 때 마다 사진을 웹 서버에 저장하고, 데이터베이스에는 해당 와인의 바코드 번호와 웹 서버에 담겨있는 사진의 URL을 저장한다. 또한 와인에 대한 평가가 이루어질 시 데이터베이스에 만족도,

가격, 마신 상황 등의 평가 정보가 저장되고 웹 서버에서 알고리즘을 통해 한 와인에 대한 평가가 추가될 때 마다 해당 와인의 가장 마시기 좋은 최적 상황을 새로 추출해낸다. 최적 상황은 와인의 정보로 데이터베이스에 저장한다. 사용자는 이를 계속해서 와인 정보로 받아볼 수 있다.

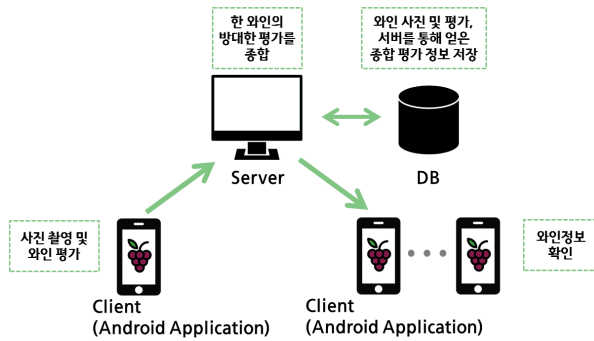


Fig. 3. System Architecture

### 3.2 데이터베이스 설계

Fig. 4는 와인추천 어플리케이션의 데이터베이스 테이블이다. 테이블은 사용자정보를 나타내는 userinfo, 사용자별 와인평가 테이블인 evaluatewine, 사용자별 와인리스트인 mywinelist, 와인의 정보를 저장하는 wineinfo, 정보수정 요청사항 테이블인 complainlist가 있다.

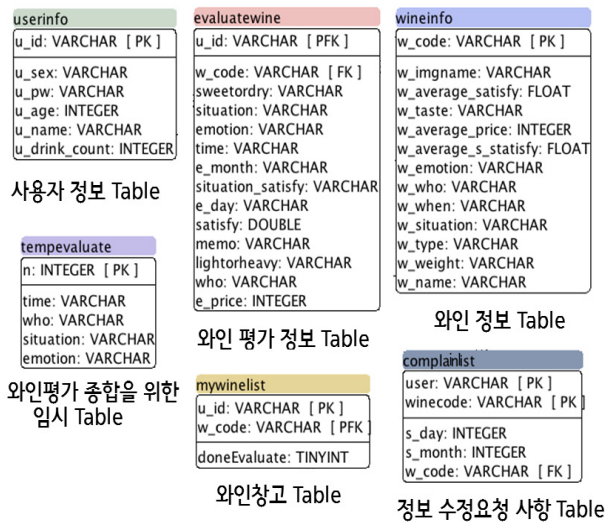


Fig. 4. Database Table

## 4. 와인 추천 어플리케이션 설계

본 장에서는 제안하는 어플리케이션의 기능에 대해서 소개한다. 3.1절에서는 바코드 인식을 이용한 와인 검색에 대해 설명하고, 3.2절은 자신만의 와인 다이어리 기능, 3.3절에서는 다양한 상황에 대한 와인을 추천받는 기능, 3.4절에서는 나만의 와인 리스트, 메모 기능에 대해 설명한다.

### 4.1 바코드 인식을 이용한 와인 검색

기존의 와인정보를 제공하는 서비스의 검색방법은 와인명을 직접치거나 와인의 라벨인식을 이용하는 방법 등이 있다. 보통의 경우 와인명을 직접 검색하기란 와인명의 길이가 길고 외국어라는 점에서 굉장히 번거롭고 불편할 수 있다. 또한 와인라벨 인식은 라벨인식 자체가 안 되는 경우도 존재 하며[6] 인식을 한다 해도 유사한 이름의 와인일 경우 사용자가 찾는 정확한 와인을 식별해내는 데는 한계가 있다.

본 논문에서 제안하는 어플리케이션은 바코드 인식 기능을 통해 사용자가 최대한 빠르고 간편한 방법으로 해당 와인의 정보를 검색할 수 있도록 한다. 단순히 바코드 스캔 화면에 바코드를 인식시키는 것으로 검색의 과정을 단순화할 수 있다. 바코드 인식을 통해 얻어낸 와인 별 바코드 넘버로 질의검색을 하게 된다. 이는 다른 와인 정보와는 독립적이고 한 와인의 고유한 값이므로 사용자가 원하는 와인을 정확하게 검색해 낼 수 있다.

### 4.2 와인 평가 및 와인정보 취합

와인정보는 기존의 형식적이고 어렵게 느껴지는 정보가 아닌 사용자들의 집단지성을 이용하는 형식의 정보를 제공한다. 집단지성을 이용한 정보 제공 방법은 다음과 같다. 첫째, 와인을 마신 사용자들에게 와인에 대한 종합만족도, 상황 및 감정 데이터, 상황만족도, 맛에 대한 정보 등을 평가 받는다. 둘째, 이를 토대로 수집된 여러 명의 평가데이터들은 종합 알고리즘을 통해서 와인에 대한 대표 정보로 수렴된다.

와인 평가 지표는 만족도, 마신 상황, 마신 상황에 대한 만족도, 단맛 정도, 무게감 정도, 자신만의 평가 또는 느낌으로 구성한다. 만족도는 별점으로 화면에서 드래그를 통해 입력할 수 있고, 상황과 그 외 평가요소는 여러 선택지 중에 하나를 선택할 수 있다. 와인 평가를 마치면 시스템적으로는 해당 와인에 대한 평가가 저장되며 와인을 마시기 좋은 최적 상황을 추출해내는 종합 과정이 진행된다. 또한, 사용자에게는 와인의 정보에 자신의 평가도 적용되었다는 것을 알린다. 이를 통해 사용자는 자신의 평가에 대한 신중함과 즉각적인 변화에 즐거운 호기심을 가질 수 있을 것으로 기대한다.

모든 와인의 평가들을 바탕으로 해당 와인을 마시기 좋은 가장 최적의 상황을 뽑아낸다. 평가 시에 입력받은 상황과 해당 상황에 대한 만족도를 토대로 4.3절에서 설명할 ‘와인 최적 상황 추출’ 과정을 거쳐 와인 정보 화면에서 이를 확인할 수 있게 제공한다.

### 4.3 다양한 상황에 대한 와인 추천

빅 데이터를 기반으로 하여 사용자 개인에 맞는 맞춤형 와인 추천 기능을 제공한다. 와인 추천 서비스는 크게 4가지로 구분한다.

첫 번째 기능은 Top-k를 이용한 종합 랭킹 서비스이다. 을 제공한다. 데이터베이스에 존재하는 와인 평가 데이터를 기준으로 와인에 대한 평점을 정렬한 후, 가장 평점이 높은 와인 순서대로 사용자에게 보여준다. 사용자는 Top-k의 와인 목록을 통해 가장 선호도가 높은 와인의 종류를 한눈에 파악할 수 있다. 이는 실시간으로 변동되며, 사용자를 지속적으로 확보하는 것을 목적으로 한다.

두 번째 기능은 나이, 성별, 시간대로 구분되는 사용자의 목적과 조건에 맞는 와인 추천 서비스이다. 위 Top-k의 종합 랭킹과 같은 알고리즘을 적용하여 나이, 성별, 시간대로 데이터를 분류하여 그룹화 한다. 사용자는 조건에 맞는 전체적인 와인 정보를 선호도가 높은 순으로 볼 수 있다.

세 번째 기능은 언제, 어디서, 누구와 같은 사용자의 상황과 기분을 고려하여 와인을 추천하는 서비스이다. 5개 질문에 응답하는 형식으로 UI를 제공하여 보다 사용하기 쉽고 간편한 서비스를 제공한다. 사용자는 상황별 추천 기능을 통해 개인에게 맞는 와인을 보다 정확하게 추천받을 수 있으며 이는 기존 와인에 대한 정보만을 제공하는 경쟁사와 다른 차별성으로 경쟁력을 높일 수 있다고 기대한다.

마지막 기능은 3일 내 가장 많이 찾은 와인 정보를 제공하는 오늘의 와인 추천 서비스이다. 3일 내 가장 많이 평가된 와인 리스트를 추출하여 사용자의 평점 순으로 정렬 후, 가장 높은 평점을 가진 Top 3의 와인 리스트를 보여준다.

#### 4.4 나만의 와인 리스트 및 메모

기존의 메모 어플리케이션을 사용할 경우, 와인정보에 대한 메모를 할 때 와인에 대한 모든 정보를 일일이 입력하기 때문에 불편하다. 제안하는 어플리케이션은 내가 마셨던 와인에 대한 정보를 기록할 수 있도록 간단한 과정의 와인 리스트와 나만의 메모기능을 제공한다.

와인 리스트 기능은 와인에 대한 정보를 저장할 수 있는 기능으로써 와인 평가이다. 이는 와인 리스트 기능의 쓰임을 다양화 할 수 있는 요소로, 사용자에게 따라 내가 마신 와인에 대한 기록으로 활용할 뿐만 아니라 앞으로 마셔보고 싶은 와인리스트로 사용할 수 있다. 와인리스트기능 내에서는 평가가 완료된 와인과 평가되지 않은 와인을 사용자가 시각적으로 쉽게 구분이 가능하다.

와인에 대해 평가를 했을 경우에도 자동으로 와인 리스트에 추가가 되며, 이때는 사용자가 평가한 나의 와인 평가도 함께 저장된다. 또한 평가과정에서 개인적인 메모남기기가 가능하기 때문에 직접 나의 와인 평가와 메모를 기록함으로써 기본적으로 되고 오디는 와인정보와 개인의 와인평가에 대한 정보를 비교할 수 있다.

### 5. 와인 추천 어플리케이션 구현

본 장에서는 제안하는 와인 추천 어플리케이션의 설계 및 구현에 대해서 소개한다. 실제 시스템 구현은 APM(Apache, Php, MySQL)으로 되었다. 4.1절에서는 바코드 인식을 이용

한 와인 검색이 이루어지는 과정에 대해서 설명하고, 4.2절은 자신만의 와인 다이어리 기능, 4.3절에서는 다양한 상황에 대한 와인을 추천받는 기능, 4.4절에서는 나만의 와인 리스트, 메모 기능이 진행되는 과정에 대해서 설명한다. 4.5절에서는 본 설계의 시스템 구성도, 서버 설계와 데이터베이스 설계에 대해 설명한다.

#### 5.1 바코드 인식을 이용한 와인 검색

본 절에서는 3.1절에서 설명한 바코드 인식을 이용한 와인 검색 기능에 대한 알고리즘 적용 방법을 소개한다. 바코드 인식을 이용한 와인 검색의 과정은 바코드 인식, 바코드 넘버추출, 데이터베이스 와인정보 유무검색, 와인정보가 있다면 와인정보 출력, 와인정보가 없다면 새로운 와인에 대한 사용자의 데이터를 출력한다.

Fig. 5는 위와 같은 과정을 종합한 바코드를 통한 와인 정보 출력 알고리즘이다. 바코드 스캔을 통해 얻어진 바코드 넘버(와인코드)를 이용하여 데이터베이스의 *wineinfo* 테이블에서 질의를 한다. Fig. 4 스키마의 *w\_code* 어트리뷰트 중 바코드 넘버와 일치하는 것이 있는지 질의하고, 결과 값으로 바코드 넘버와 일치하는 *w\_code*가 존재한다면 이미 와인 데이터가 있는 것이고, *w\_code*가 존재하지 않는다면 사용자가 검색한 와인에 대한 정보가 없는 것이다.

해당하는 *w\_code*가 존재한다면 모든 와인정보들을 불러와 와인 정보 창에 출력해준다. 해당하는 *w\_code*가 존재하지 않았다면 사용자에게 와인정보 입력을 유도하고, 사용자가 검색한 바코드 넘버와 사용자 입력한 와인의 사진, 와인명, 와인 종류 등을 *wineinfo*의 새로운 튜플로 삽입시킨다. 새로 등록된 와인의 경우, 사용자들의 평가가 종합되어 와인에 대한 다른 정보들이 생성될 기반 튜플이 된다.

---

#### Algorithm : Wine Information Printing

---

**Input.** *Barcode* : Wine's unique number

**Output.** *Show wine information*

**Algorithm:**

- 1: Get information of the barcode.
  - 2: **IF** there is a Barcode in DB.  
Show wine information.
  - 3: **ELSE** Store information of new wine from user. Send new wine's information to DB. Show wine information.
- 

Fig. 5. Wine Information Printing Algorithm

#### 5.2 와인 평가 및 와인정보 취합

본 절에서는 3.2절에서 언급한 와인을 평가하고 와인의 정보를 종합하는 기능에 대한 알고리즘 적용 방법을 소개한다. 먼저 와인 평가는 와인에 대한 사용자들의 평가를 모아 놓는 *evaluatewine* 테이블에 해당 와인의 바코드 값인 *w\_code*와 *u\_id*를 가지고 평가 정보를 구분하여 저장하게

된다. 이때, 사용자 정보를 기록하는 *userinfo* 테이블의 사용자가 마신 와인의 수를 기록하는 *u\_drink\_count* 어트리뷰트의 값이 1 증가된다. 또한, 사용자의 '내 와인 창고'에 추가되어야 하기 때문에 와인창고 정보를 가지는 *mywinelist* 테이블에 *u\_id*, *w\_code*와 함께 *doneEvaluate* 어트리뷰트로 평가를 완료한 와인이라는 정보를 저장한다.

다음으로, 한 와인에 대해 마시기 좋은 최적의 상황을 제시하기 위해서 해당 와인의 대한 사용자들의 많은 평가를 종합하는 알고리즘이 필요하다. 첫 번째는, 와인의 대한 평가는 마신 상황과 더불어 마신 상황에 얼마나 잘 맞는 와인 이었는지를 별점으로 0.5점부터 5점까지 0.5점 단위로 나누어 입력받는다. 그렇기 때문에 한 와인의 평가 중 0.5점의 상황과 5점의 상황은 다르게 적용되어야 한다. 두 번째는, 상황은 언제, 누구랑, 무엇을 할 때, 어떤 감정으로 마셨는지를 기록하기 때문에 4가지 마다의 속성 값들을 어떤 식으로 종합할 수 있는가에 대한 것이다. 따라서 2가지를 해결하기 위한 '최적 상황 추출 알고리즘'을 설명한다.

1) 상황 만족도 점수대 별 대표 상황 문장 도출

본 절에서는 4.2절의 와인정보를 종합하는 데 사용한 '최적 상황 추출 알고리즘'의 첫 단계인 '상황 만족도의 점수대 별 대표 상황 문장 도출' 과정을 소개한다.

Table 1은 한 와인에 대한 평가를 가정하여 10개의 튜플을 작성한 것이다. 상황 만족도가 5점인 경우에 대표 문장이 어떻게 도출되는지 보여준다. 문장 내용을 결정짓는 4가지의 속성들 중 가장 두드러지는 값을 가진 속성을 해당 속성 값으로 고정하는 것이다. 예를 들어 설명하면, 현재 '언제' 속성에는 '아침'이 4번 등장하고, '누구랑' 속성에는 '연인'과 '가' 6번 등장하고, '어느 상황에' 속성에는 '밥 먹으며'가 5번 등장하고, '어떨 때' 속성에는 '심심할 때'가 4번 등장한다. 이 중 가장 큰 값이 등장한 것은 '연인'과이다. 따라서 '누구랑' 속성에 '연인'과를 고정하게 된다. 이때 5점의 대표 문장 중 한 가지가 채워진 것이다. 이는 데이터베이스에서 *evaluatewine* 테이블에서 바코드로 얻어진 해당 와인의 바코드 번호를 통해 일치하는 *w\_code*의 튜플에 저장된다.

Table 1. An Example of Situation Satisfaction

상황 만족도	언제	누구랑	어느 상황에	어떨 때
5	저녁	연인	밥 먹으며	행복할 때
5	아침	혼자서	밥 먹으며	심심할 때
5	저녁	연인	밥 먹으며	행복할 때
5	점심	연인	데이트하며	행복할 때
5	아침	연인	밥 먹으며	심심할 때
5	점심	연인	데이트하며	심심할 때
5	점심	친구와	파티하며	신날 때
5	아침	혼자서	자기 전에	외로울 때
5	아침	가족과	밥 먹으며	심심할 때
5	저녁	연인	기념일 날	화날 때

그 이후로 고정된 '연인'과라는 값을 가진 튜플만을 다시 본다. 그러면 1, 3, 4, 5, 6, 10번째 튜플만 다시 살펴보는 것이다. 여기서 앞서 설명한대로 가장 두드러지는 값을 고정하게 된다. 결과적으로 '어느 상황에' 속성에 '밥 먹으며'가 5점의 대표 문장 중 한 가지를 또 채워지게 된다. 이러한 방식으로 반복하면 순차적으로 '언제' 속성에 '저녁'이, '어떨 때' 속성에 '행복할 때'가 채워지게 된다.

이렇게 5점의 대표 문장이 만들어지고 같은 방식으로 0.5점 단위로 0.5점부터 5점까지 대표 문장이 만들어지므로 총 10개의 대표 문장이 완성된다. 이 10개의 문장 값들은 데이터베이스의 점수대별 대표 문장을 임시적으로 저장해놓는 *tempevaluate* 테이블에 저장된다. 점수를 가지고 있는 n(n의 값은 0.5-5까지 총 10개로 고정), '언제' 속성을 기록한 *time*, '누구랑' 속성을 기록한 *who*, '어느 상황에' 속성을 기록한 *situation*, '어떨 때'를 기록한 *emotion*에 기록된다.

2) 차등 점수 부여 및 합산을 통한 최종 상황 문장 도출

'최적 상황 추출 알고리즘'의 두 번째 단계인 점수대 별로 만들어진 대표 상황 문장을 토대로 차등 점수를 부여 및 합산을 통해 최종 상황 문장을 도출하는 과정을 소개한다.

Table 2는 10개의 점수대 별 대표 문장들이 만들어 결과를 보여준다. 대표 문장을 만들었으나 0.5점의 대표 문장의 의미와 5점의 대표 문장의 의미는 다르다. 0.5점의 경우 해당 상황에 마셨을 때 아주 별로였다는 의미이고, 5점의 경우 해당 상황에 아주 좋았다는 의미이기 때문에 다르게 표현되어야 한다. 따라서 해당 문장 별로 차등 점수를 부여해 합산하여 가장 높은 점수를 갖는 값을 최종 대표 문장으로 정리한다. 예를 들어서, '언제' 속성의 '아침'의 총 점수는 5점, 4.5점, 3점, 1.5점, 1점을 모두 더한 값인 15점이다.

Table 2. An Example of Statement for each Score

상황 만족도	언제	누구랑	어느 상황에	어떨 때
5	아침	연인	밥 먹으며	행복할 때
4.5	아침	혼자서	밥 먹으며	심심할 때
4	저녁	연인	밥 먹으며	행복할 때
3.5	점심	연인	데이트하며	행복할 때
3	아침	연인	밥 먹으며	심심할 때
2.5	점심	연인	데이트하며	심심할 때
2	점심	친구와	파티하며	신날 때
1.5	아침	혼자서	자기 전에	외로울 때
1	아침	가족과	밥 먹으며	심심할 때
0.5	저녁	연인	기념일 날	화날 때

이러한 방식으로 각 속성 별로 가장 점수가 높은 것을 추출해보면 '저녁'이 15점으로, '연인'과가 15점으로, '밥 먹으며'가 19점으로, '행복할 때'가 14.5점으로 최종 도출된다. 이러한 과정을 거쳐 *wineinfo* 테이블에 해당 와인의 *w\_code* 값을 갖는 *time*, *who*, *situation*, *emotion* 어트리뷰트에 값

이 채워지게 된다. 최종적으로 와인의 마시기 좋은 최적 상황이 결정된 후에는 앞서 임시적으로 점수대별 대표 문장을 기록하기 위해서 사용한 *tempevaluate* 테이블의 *time*, *who*, *situation*, *emotion*의 값을 계속해서 최적 상황 추출에 사용하기 위해 모두 0으로 바꾼다.

5.3 다양한 상황에 대한 와인 추천

본 절에서는 3.3절에서 언급한 상황별 와인 추천 기능에 대한 알고리즘 적용 방법을 소개한다. 알고리즘의 수행 단계는 첫째, Top-k 종합 랭킹, 둘째, 와인매칭기능, 셋째, 와인추천기능이다.

첫째, Top-k 종합 랭킹은 Fig. 6과 같이 데이터베이스의 *wineinfo* 스키마의 *w\_average\_satisfy* 어트리뷰트를 기준으로 데이터 목록을 추출한다. *evaluatewine* 스키마의 *satisfy* 어트리뷰트에 사용자로부터 입력받은 평점을 1-5점안에 해당하는 값으로 치환하여 저장한다. 클라이언트 부분에서 와인에 대한 평가 이벤트가 발생할 때마다 서버의 알고리즘을 실시간으로 적용하여 와인에 대한 종합평점이 추출되도록 한다. *evaluatewine*의 복합 *key*를 구성하는 값 중 하나인 *w\_code*을 기준으로 튜플들을 뽑아 각 와인에 해당하는 종합 평점을 구한다. 구한 종합 평점은 *wineinfo* 스키마의 *w\_average\_satisfy* 에 저장된다. 클라이언트에서는 서버에서 계산된 *w\_average\_satisfy*의 값이 높은 순서대로 리스트뷰에 출력된다. 사용자는 리스트뷰의 목록을 보며 평점이 높은 순서대로 와인에 대한 정보를 얻을 수 있다.

**Algorithm : Top-k Condition Recommendation**

**Input.** *Score : Wine's satisfy score*  
*Atmosphere : When, who, what, emotion*  
**Output.** Show Top-k wine list  
**Algorithm:**

- 1: // Using Top-k, select whole wine list.
- 2: **SELECT** wine list
- 3: **ORDER BY** satisfy and show wine list information

Fig. 6. Top-k Condition Recommendation Algorithm

둘째, 사용자의 상황과 기분별에 따른 와인 매칭 기능은 Fig. 7과 같이 *wineinfo* 테이블의 *w\_when*, *w\_who*, *w\_what*, *w\_emotion* 어트리뷰트에 맞는 와인 목록들을 추출한다. 사용자가 응답한 상황, 기분의 값을 데이터베이스의 위에 언급한 어트리뷰트와 비교하여 해당하는 와인 목록을 매칭한다.

마지막으로 3일 내 가장 많이 찾은 와인 정보를 제공하는 오늘의 와인 기능은 Fig. 8과 같이 평가받은 날짜인 *evaluatewine* 스키마의 *time* 어트리뷰트를 기준으로 추출한다. *time*의 값이 오늘 날짜와 3일 내에 평가된 와인 리스트를 추출한 후, 각 와인의 *key*값으로 그룹화 하여 평가된 와인 튜플 수를 카운트한다. 그 중 가장 많이 평가된 Top 3의 와인 키 값을 추출

하여 해당 *wineinfo*의 *satisfy*를 기준으로 정렬한 다음, Top 3의 와인 리스트를 보여지게 된다.

**Algorithm : Atmosphere Matching**

**Input.** *Emotion : User's emotion*  
*User's situation : When, Who, What*  
**Output.** Top-k wine list  
**Algorithm:**

- 1: // Get information of user's emotion and situation
- 2: **SELECT** Wine list which have user's emotion and situation
- 3: **ORDER BY** Satisfy // show wine list information

Fig. 7. Atmosphere Matching Algorithm

**Algorithm : Today's Wine Recommendation**

**Input.** *Score : Wine's satisfy score*  
*Date : When user's evaluate time*  
**Output.** Show the most evaluated 3 wine information in 3 days from today  
**Algorithm:**

- 1: // Get today's date
- 2: **SELECT** Whole wine list in 3 days From today
- 3: **COUNT** Those wine list **GROUP BY** Key(*w\_code*) **ORDER BY** desc.
- 4: // Show selected wine list **ORDER BY** Evaluated mount in 3 days From today

Fig. 8. Today's wine recommendation Algorithm

5.4 나만의 와인 리스트 및 메모 기능

본 절에서는 3.4에서 언급했던 나만의 와인 리스트, 메모 기능에 대한 알고리즘을 설명한다. Fig. 9는 나의 와인 리스트에 와인 담기 알고리즘을 보여준다.

와인정보를 내 와인 리스트에 담는 것은 아주 간단한 과정을 통해 이루어진다. 와인정보 화면에서 내 와인창고에 담기 버튼을 이용해 와인리스트에 와인을 담을 수 있다. 이 과정은 와인정보 화면에 출력되는 와인 정보에 대한 와인코드를 와인리스트 추가하는 php로 전달하고 php에서 데이터베이스 *myWineList*와 사용자의 아이디를 복합키로 이용해서 사용자의 와인리스트에 이미 담겨있는 와인인지 아닌지 검사한다. 그 후에 담겨있지 않은 와인이라면 전달된 와인코드와 사용자의 아이디를 *myWineList*의 새로운 튜플로 추가한다. 이때 이미 담겨 있는 와인이라면 사용자에게 이미 담겨진 와인이라는 것을 알려준다. 와인정보 화면에서 사용자가 바로 평가하기로 들어가서 와인에 대한 평가를 하면 자동으로 내 와인 리스트에 선택한 와인이 저장된다.

와인리스트에서 내가 담은 와인에 대한 정보를 확인하는 과정은 다음과 같다. 아래의 Fig. 9를 통해서도 알고리즘을 확인 할 수 있다. 먼저 유저의 아이디를 통해 데이터베이스의 MyWineList에서 w\_code를 검색한다. 이때 검색된 w\_code가 포함된 튜플들의 개수를 이용해서 내 와인리스트의 와인 개수를 계산한다. 그리고 doneEvaluate 어트리뷰트의 값이 1이면 평가여부에 ‘평가완료’를, 0이면 ‘평가안됨’을 화면에 출력해준다. 검색된 w\_code들로 WineInfo 테이블에서 wine의 사진(w\_imgname)과 와인 명(w\_name) 또한 추출해서 화면에 출력해준다. 와인리스트의 각 리스트들을 클릭하면 와인 정보 화면으로 이동되어 와인에 대한 정보, 내 와인평가, 나만의 메모 기능들을 볼 수 있다.

**Algorithm : Add My Wine List Algorithm**

**Input.** User's ID

Barcode number of wine

**Output.** IsDoneEvaluate, Wine's name, Wine's picture info

**Algorithm:**

- 1: **IF** Barcode number of wine was enrolled, result is previous enrolled
- 2: **ELSE** Select isDoneEvaluate value in the MyWineList table and using the barcode number in the wineinfo, select wine's name, wine's picture info

Fig. 9. Add my wine list algorithm

**6. 구현 결과**

본 장에서는 제안하는 어플리케이션의 구현을 통한 기능별 구현 결과 화면을 소개한다.

**6.1 바코드 인식을 이용한 와인 검색**

Fig. 10은 제안하는 어플리케이션의 메인화면이다. 메인화면에서 우측하단의 바코드 검색 버튼을 클릭하면 Fig. 11의 바코드 스캔화면으로 전환된다. 바코드 스캔화면에서 검색하고자하는 와인의 바코드를 인식시킨다. 바코드 검색결과가 없을 경우 최초 검색자에게 Fig. 12 처럼 와인의 사진, 와인명, 와인 종류를 선택하는 창으로 전환되는 것을 볼 수 있다. 이 화면에서 입력완료버튼을 누르면 이 입력한 와인을 기본으로 한 와인정보 화면으로 전환된다. 바코드 검색결과 와인에 대한 정보가 있을 경우에는 Fig. 13과 같이 와인정보 화면에 와인에 대한 정보를 출력하는 창으로 전환된다.

**6.2 와인 평가 및 와인 정보 출력 화면**

Fig. 14에서는 내가 마신 와인에 대해 평가하는 평가 화면이다. 와인에 대한 종합만족도, 구입가격, 와인을 마신 상황과 감정정보(언제, 누구랑, 어느 상황에, 어떨 때), 마신상황에 대한 상황만족도, 와인의 맛 정보(당도, 무게감) 등을 입력받는다. Fig. 15에서는 사용자의 와인 평가들을 종합하

여 와인에 대한 정보들을 제공해주는 화면이다. 와인 명, 와인 사진, 종합 만족도, 마시기 좋은 상황, 와인 종류, 무게감, 당도, 평균 가격 등을 확인할 수 있으며 내가 평가한 와인인 경우 내가 평가한 와인의 정보도 함께 볼 수 있다.

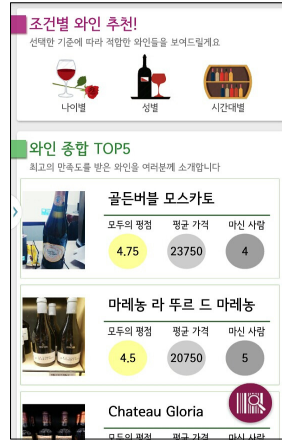


Fig. 10. Main Screen

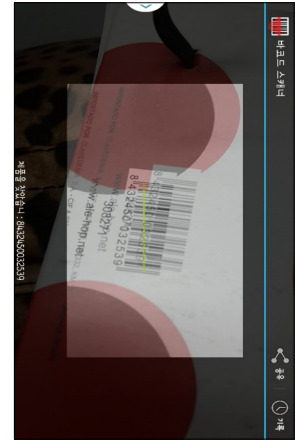


Fig. 11. Barcode Scanning



Fig. 12. Wine Information Input Screen



Fig. 13. Wine Information Screen

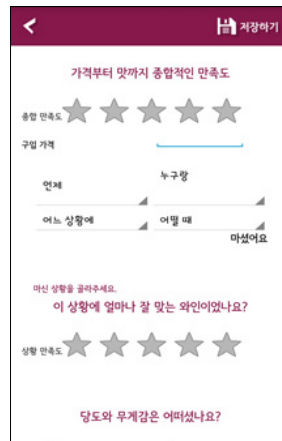


Fig. 14. Wine Evaluation Screen



Fig. 15. Wine Information Screen

6.3 다양한 상황에 대한 와인 추천 기능

Fig. 16은 메인화면에서 제공하는 와인 종합 Top5 랭킹이다. 이는 세 명 이상의 평가를 받은 와인들의 종합 만족도가 높은 와인 순으로 보여 준다. Fig. 17, Fig. 18과 Fig. 19는 조건별로 와인의 랭킹을 보여주는 화면들이다. Fig. 17은 연령대별로 어떤 와인을 선호하는지를 알 수 있다. 마찬가지로 Fig. 18은 성별로 선호하는 와인을, Fig. 19는 시간대별로 어떤 와인들이 인기 있는지에 대한 랭킹을 제공한다.

Fig. 20과 Fig. 21은 와인 5고개를 통해 사용자 상황에 맞춤형 와인을 추천해주는 추천기능의 일부 화면으로 마실 시

간대를 물어본다. 이밖에도 함께 마실 사람, 마시는 상황, 마시게 될 기분, 원하는 와인의 맛 등을 물어보고 이에 맞는 와인을 찾아준다.

6.4 와인다이어리 기능

Fig. 22는 내가 마신 와인들을 따로 저장해 둘 수 있는 와인창고의 화면이다. 와인정보가 검색되면 와인창고에 추가하기 버튼을 누르거나, 평가를 하면 추가할 수 있다. Fig. 23은 내가 남긴 와인에 대한 메모를 확인하는 와인정보창의 화면이다. 와인창고와 메모기능을 이용해서 내가 마신 와인에 대한 기록을 남기고 쉽게 확인할 수 있다.



Fig. 16. Wine Top5 Screen



Fig. 17. Wine Ranking per Age Screen



Fig. 18. Wine Ranking per Gender Screen



Fig. 19. Wine Ranking per Time Screen

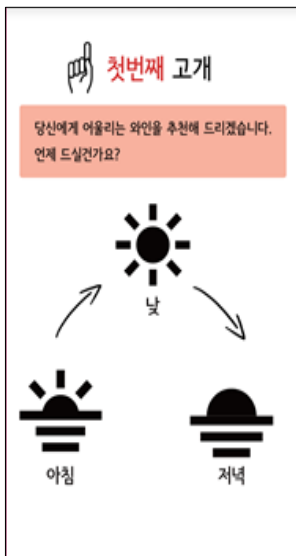


Fig. 20. Wine Five-Question Screen



Fig. 21. Wine Five-Question Result Screen



Fig. 22. Wine List

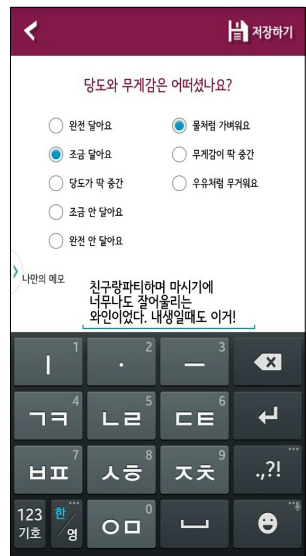


Fig. 23. My Evaluation & Memo



## 7. 결 론

본 논문에서는 앙트러프러너십 과정의 예비 창업자들에게 기반 기술 지원 측면에서 와인 사업을 위한 데이터를 제공하는 모바일 콘텐츠 어플리케이션을 구현한다. 또한 창업자 및 고객들에게 맞춤형 와인을 추천 해주는 기능을 통해 예비 창업자들에게 고객의 니즈를 파악할 수 있는 지식베이스를 제공하기 위해 방대한 양의 와인에 대한 정보를 제공하고 선택 지표를 마련하는 시스템을 제안하였다.

본 논문은 와인에 대한 형식적인 정보만을 제공하는 기존 와인 관련 서비스와는 달리 다양한 상황과 분위기, 사용자를 고려해 와인 선택에 도움이 되는 지표를 제공하며 더욱 사용자들의 선호도를 높인다. 또한 사용자가 직접 마신 와인에 대한 평가를 종합하여 소비자의 평가를 반영한 와인 정보를 제공하고 바코드 인식이라는 검색 기능을 통해 쉽고 간편한 접근성을 제공하며 어플리케이션의 효율성을 갖춘다. 마지막으로 와인 다이어리 기능을 통해 개인적으로 마신 와인에 대한 기록을 할 수 있어 사용자가 어플리케이션을 더욱 편리하게 사용할 수 있도록 한다.

향후 연구에서는 다른 유저들과의 소통의 장이 될 수 있도록 공유 및 댓글 기능 등의 커뮤니케이션 서비스를 구현하고자 한다. 또한 와인 검색방법의 다양화를 통해 와인 검색의 편리성을 향상시켜 와인사업을 준비 하는 예비 창업자들을 지원하고자 한다.

## References

- [1] J. W. Park, "The Level of Specialization of Wine Consumers Wine Market, According to the Level of Characterization," *Northeast Asia Tourism Research*, Vol.10, No.4, pp.145-163, 2014.
- [2] H. S. Park, "The Competitiveness of the Korean Wine Industry with World Wine Market Analysis," *Journal of Commodity Science and Technology*, Vol.27, No.1, pp.93-107, 2009.
- [3] J. M. Jung, H. J. Yang, S. H. Kim, G. S. Lee, and S. H. Kim, "Wine Label Recognition System using Image Similarity," *The Journal of the Korea Contents Association*, Vol.11, No.5, pp.125-137, 2011.
- [4] C. Rowena, L. Diego, C. Scott, and H. Damian, "Labelling: Wine labelling - future considerations and trends," *Wine & Viticulture Journal*, Vol.30, No.2, 2015.
- [5] M. R. Jung, "Exploring Differences of Purchase Motives and Information Source of Wine Consumers: Focused on Demographic Profiles," *Journal of Tourism and Leisure Research*, Vol.23, No.4, pp.313-327, 2011.
- [6] J. S. Lim, S. H. Kim, C. W. Lee, G. S. Lee, H. J. Yang, M. E. Lee, "Wine Label Character Recognition in Mobile Phone

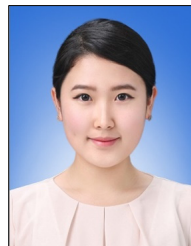
Images using a Lexicon-Driven Post-Processing," *Journal of KIISE: Computing Practices and Letters*, Vol.16, No.5, pp.546-550, 2010.

- [7] J. M. Wolfe, "Visual search in continuous, naturalistic stimuli," *Vision Research*, Vol.32, No.9, pp.1187-1195, 1994.
- [8] J. K. Park, "A Study on Application Methods of Analytical Technologies for Analyzing Big Data," M.S. Thesis, University of Sejong, pp.1-55, 2013.



### 박 시 명

e-mail : aimgseem@gmail.com  
 2012년~현 재 숙명여자대학교  
 멀티미디어학과 학사과정  
 관심분야 : 빅데이터, 안드로이드, UX/UI



### 윤 소 영

e-mail : yoonso225@gmail.com  
 2012년~현 재 숙명여자대학교  
 멀티미디어학과 학사과정  
 관심분야 : 안드로이드, 머신러닝, 빅데이터



### 서 은 비

e-mail : dmsql0722@gmail.com  
 2012년~현 재 숙명여자대학교  
 멀티미디어학과 학사과정  
 관심분야 : 서버, 네트워크, 네이티브 언어



### 손 중 서

e-mail : jsson@sm.ac.kr  
 2007년 중앙대학교 창업학(석사)  
 2013년 중앙대학교 창업학(박사수료)  
 2007년~2009년 하나창업투자 상무  
 2007년~2010년 IFS Korea 대표  
 2012년~현 재 숙명여자대학교 글로벌

서비스학부 앙트러프러너십전공 조교수  
 관심분야 : 창업, 공공데이터



**박 소 현**

e-mail : shpark@sm.ac.kr  
2014년 숙명여자대학교 피아노과 학사  
2014년~현재 숙명여자대학교  
멀티미디어과학과 석사과정  
관심분야: 빅데이터, HCI, 영상처리,  
멀티미디어 콘텐츠, 음악 교육



**박 영 호**

e-mail : yhpark@sm.ac.kr  
1992년 동국대학교 컴퓨터공학과(공학석사)  
2005년 한국과학기술원 진산학과(공학박사)  
1993년~1999년 한국전자통신연구원 교환  
전송연구단 선임연구원  
2005년~2006년 한국과학기술원 첨단정보  
기술연구센터 연구원  
2005년~2006년 동국대학교 컴퓨터멀티미디어학과 겸임교수  
2006년~현재 숙명여자대학교 멀티미디어과학과 교수  
관심분야: 데이터베이스, XML, IR(정보검색), 멀티미디어 데이터  
베이스, Bio정보공학, 영상미디어, 예술&공학인터페이스,  
데이터베이스 관리시스템, 머신러닝, 빅데이터, 데이터  
분석, Telecommunication System