
사용자 편의성과 시스템 효율성을 지원하는 대리운전 시스템

고정국*

A proxy driving system supporting user convenience and system efficiency

Jeong-Gook Koh*

요 약

본 논문에서는 스마트폰을 활용하여 고객에게는 편리한 대리운전 신청 기능을, 대리기사에게는 고객의 위치정보를 제공하여 신속한 서비스를 제공할 수 있는 3-tier 구조의 대리운전 시스템을 설계하고 구현하였다. 구현된 대리운전 시스템은 기존의 전화신청 방식과 스마트폰을 활용한 온라인 주문 방식을 모두 지원하며, 시스템 성능 분석을 통해 온라인 주문 비율이 높을수록 대리운전 접수와 처리의 자동화를 통해 콜센터 상담원의 업무량 감소와 대리운전 신청 처리시간 단축 효과를 제공함을 확인하였다. 대리운전 업체 입장에서는 구현된 시스템을 활용하여 상담원의 증원을 최소화하고 서비스 품질을 개선하여 고객 만족도를 향상시키고 비용 절감과 시장 점유율 확대 효과도 거둘 수 있다.

ABSTRACT

This paper describes the design and implementation of a 3-tiered proxy driving system utilizing smartphones. This system provides customers with a convenient request facility, and enables proxy drivers to provide a prompt service based on customer's location information. Implemented system supports two methods such as the existing phone-call and the online requesting method using smartphones. Performance analysis shows that the higher the ratio of online requests, the greater the effect of decreasing of phone jockey's workloads and the reduction of request processing time through the automation of receipt and processing. From a proxy driver company's standpoint, this system minimizes the increase of phone jockeys, and improves customer satisfaction through good service quality. This system also helps the company reduce the cost and increase the market share.

키워드

대리운전, 대리기사, 스마트폰, 3-tier 구조

Key word

Proxy driving, proxy driver, smartphone, 3-tier architecture

* 정회원 : 동명대학교 컴퓨터공학과(교신저자, jgkoh@tu.ac.kr)

접수일자 : 2012. 04. 02

심사완료일자 : 2012. 05. 08

Open Access <http://dx.doi.org/10.6109/jkiice.2012.16.8.1725>

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서 론

IT 기술의 발전에 따라 기업의 경쟁 기준이 원가 중심의 차별화 전략에서 고객 중심의 가치창조 방식으로 패러다임이 변하고 있다. 고객 중심 측면에서 신규 고객의 확보와 유지는 기업의 경쟁력 확보에 매우 중요하며, 고객들의 만족도 향상을 위해서는 요구사항을 철저히 분석할 필요가 있다. 대리운전은 고객관리 활동의 대표적인 사례로서, 고객의 대리운전 신청을 전화로 접수하고 처리하던 단순한 수준에서 고객 만족을 위한 중요한 채널로 자리잡고 있다[1].

차량 보급률의 증가에 따라 음주 운전으로 인한 교통사고도 증가함에 따라 음주 운전의 위험과 폐해에 대한 사회적 공감대가 확산되고 교통 법률도 강화되고 있다. 이러한 상황에서 음주 운전의 예방책으로 대리운전이 대두되면서 대리운전 업체는 전국적으로 9천여개, 대리기사는 15만 명에 달하는 치열한 경쟁 상황에 직면하고 있다[2].

따라서 대리운전 업체들은 고객들의 서비스 만족도 향상과 시장 점유율 확대를 위해 콜센터의 상담원 증원과 관련 인프라 확충이 필요하지만 대리운전 신청이 특정 시간대에 집중되는 대리운전의 특성과 업체의 영세성을 감안할 때 상담원 증원과 인프라 확충을 최소화하면서 피크 시간대에도 양질의 서비스를 제공할 수 있는 방법이 요구된다[3].

한편, 스마트폰 시장 선점을 위한 휴대폰 생산업체들의 치열한 경쟁과 사용자들의 관심이 증대되면서 스마트폰이 빠른 속도로 확산되고 있다.

본 논문에서는 위치기반 서비스를 제공하는 스마트폰을 활용하여 고객에게는 편리한 대리운전 신청 기능을, 대리기사에게는 고객의 위치 정보를 제공하여 신속한 서비스를 가능하게 하는 3-tier 구조의 대리운전 시스템을 제안한다. 제안하는 시스템은 기존의 전화 신청 방식과 스마트폰을 활용한 신청 방식을 모두 지원하여 콜센터의 피크 시간대 업무 부하를 경감시킴으로써 상담원 증원을 최소화하면서 대리운전 접수 및 처리 시간의 단축 효과를 제공한다.

본 논문의 구조는 다음과 같다. 2장에서는 대리운전 시스템의 개발 사례와 3-tier 구조에 대해 설명한다. 3장에서는 본 논문에서 제안한 대리운전 시스템의 구조, 기

능 설계 및 구현 내역을 기술한다. 4장에서는 구현된 시스템의 성능을 분석하며, 5장에서는 본 논문의 결론 및 향후 연구방향을 제시한다.

II. 관련 연구

2.1. 대리운전 시스템의 개발 사례

대리운전 서비스의 초기에는 고객이 대리운전 업체에 직접 전화를 걸어 서비스를 신청하였다. 대리운전 업체는 통화에 의해 고객의 위치를 파악하고 대리운전이 가능한 기사를 검색한 후 고객에게 전화를 걸어 대리운전 서비스가 가능함을 알리고 대리기사의 대략적인 도착시간을 알려주었다. 이 방식은 고객이 대리운전 업체의 전화번호를 숙지해야 하는데 음주상태에서 전화번호를 기억하지 못할 경우 대리운전 서비스를 이용하기 어려운 단점이 있다.

대리운전 업체는 서비스 이용의 불편함 때문에 고객이 감소하면 수입이 줄어들며, 대리기사 수가 한정되어 있으므로 한명이 담당해야 할 지역이 넓은 경우 신속한 서비스 제공이 어려워 서비스 품질이 저하되는 문제점이 있었다[4]. 이러한 문제점을 해결하기 위해 네트워크를 통한 대리운전 신청 방법[4]과 대리기사나 고객의 위치 정보를 이용하는 방법들[5,6,7,8,9]이 개발되었고, 최근에는 스마트폰을 활용하는 방법들[10,11,12,13,14]도 등장하였다.

네트워크를 이용한 신청 방법은 고객이 대리운전업체에 전화를 걸지 않더라도 유·무선 네트워크를 이용하여 연결함으로써 대리운전 서비스를 이용할 수 있다. 고객이 대리운전업체의 전화번호를 기억하지 못해도 네트워크에 접속하면 대리운전을 편리하게 신청할 수 있다는 장점이 있다.

위치 정보를 이용한 대리운전 서비스 제공 방법은 전화국이나 이동통신사, GPS 위성을 이용하여 고객과 대리기사의 현재 위치를 파악한 후 고객으로부터 최적의 위치에 있는 대리기사를 선정하여 함으로써 고객에게 신속한 서비스를 제공할 수 있다는 장점이 있다.

스마트폰 활용 방법은 두 가지 방법의 장점을 결합한 것이며, 대리운전용 어플리케이션을 활용함으로써 사

용자는 편리함을, 대리기사는 신속한 서비스를 제공할 수 있다는 장점이 있다.

2.2. 3-tier 구조

클라이언트와 서버로 구성된 2-tier 구조는 클라이언트 수가 증가하면 네트워크 트래픽이 증가하고 서버에 과도한 부하가 유발되어 시스템 성능이 저하된다. 그리고 처리 기능이 클라이언트와 서버로 분산되어 있는 경우에는 프로그램의 변경이나 배포가 어렵다는 단점이 있다[15,16].

3-tier 구조는 2-tier 구조의 단점을 해결하기 위해 등장하였다[15,16,17,18]. 3-tier 구조는 사용자 인터페이스를 제공하는 클라이언트 계층, 서버의 기능을 수행하는 데이터 접근 계층, 그리고 클라이언트의 요청을 서버에게 전송하고 처리 결과를 클라이언트에게 제공하는 응용 계층으로 구성된다. 3-tier 구조는 확장성과 안정성, 그리고 효율 측면에서 유리하므로 대용량 서비스에 적합하며, 클라이언트의 요청이 서버에 직접 전달되지 않기 때문에 클라이언트 수가 증가하더라도 서버에 과도한 부하를 유발하지 않는다. 따라서 클라이언트 수가 많고 대용량 데이터를 다루는 분야의 시스템은 대부분 3-tier 구조를 채택하고 있다.

III. 대리운전 시스템의 설계 및 구현

3.1. 시스템 구조

대리운전 서비스의 이용 실태를 살펴보면 주간에는 거의 이용하지 않지만 야간으로 갈수록 이용 빈도가 증가하여 22시부터 새벽 2시에 정점에 이른다[3]. 따라서 대리운전 시스템을 설계할 때 시스템의 평균 부하보다 특정 시간대의 집중된 부하(peak load)를 고려해야 한다.

본 논문에서는 대리운전 신청 건수가 특정 시간대에 빠르게 증가하더라도 시스템에 과도한 부하를 유발하지 않도록 3-tier 구조를 채택하고, 기존의 전화 신청 방식과 스마트폰 어플리케이션을 통한 신청 방식을 모두 지원함으로써 콜센터의 업무 부하를 경감시켜 상담원 증원과 인프라 확충은 최소화하면서 고객들에게 양질의 서비스를 제공할 수 있는 대리운전 시스템을 제안한다.

다. 그리고 최근 스마트폰의 빠른 보급 속도를 감안하여 고객과 대리기사 모두 스마트폰을 소지하고 있다고 가정한다.

제안한 대리운전 시스템은 그림 1과 같이 고객용 어플리케이션과 대리기사용 어플리케이션, 콜 센터와 중계 서버, 그리고 데이터베이스 서버로 구성된다.

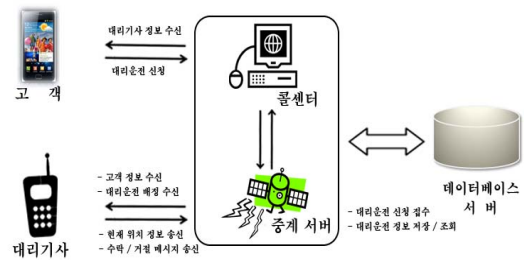


그림 1. 대리운전 시스템의 구조
Fig. 1 The architecture of a proxy driving system

3.2. 구현 환경

고객용 어플리케이션과 대리기사용 어플리케이션은 윈도우 XP SP3가 설치된 PC에서 Android SDK[19,20]가 설치된 Eclipse를 사용하여 Java로 구현하였다. 콜센터는 Eclipse와 Flex Builder 4.5 [21,22]를 사용하여 Ajax와 JSP로 구현하였다. 웹 서버로는 Apache TomCat 6.0을 사용하였고, 콜센터의 지도는 구글맵을 활용하였다. 중계 서버는 Eclipse와 BlazeDS 1.6[23]을 사용하여 Java로 구현하였다. 데이터베이스 서버로는 MySQL 5.1[24]이 사용되었고, 구현된 어플리케이션은 갤럭시 S2 스마트폰에서 기능 시험을 실시하였다.

3.3. 기능 설계 및 구현 내역

3.3.1. 고객용 어플리케이션

기존의 대리운전 서비스에서는 고객이 콜센터에 직접 전화를 걸어 현재 위치와 목적지를 알려주고 대리운전을 신청한다. 그러나 동일한 구간을 반복적으로 이용하거나 심한 음주 상태라면 이러한 절차가 번거로울 수 있다. 본 논문에서는 이러한 번거로움을 줄이기 위해 기존 방식 외에도 고객이 고객용 어플리케이션에 기본 정보(연락처, 차종, 자주 가는 곳 등)를 미리 설정한 후 “신청” 버튼만 누르면 지정된 콜센터에 정보가 전송되어 신청이 이루어지는 온라인 주문 기능도 제공한다.

고객용 어플리케이션의 시작 화면은 “신청”과 “전화 걸기” 메뉴를 제공한다. “신청” 메뉴를 선택하면 대리운전 신청 화면(그림 2(a))으로 이동하며 목적지는 설정된 값 중에서 선택한다. 설정된 기본 정보를 이용하거나 정보 변경 후 “대리운전 신청” 버튼을 누르면 신청이 완료된다. “기본 정보 설정” 기능을 이용하면 기본 정보를 설정하거나 설정 내역을 변경할 수 있다.

콜센터에서 대리기사를 선정하면 고객에게 대리기사와 요금 정보가 전송되며 출발지와 도착지, 대리기사 위치가 지도상에 그림 2(b)와 같이 표시된다. 이를 통해 대리기사와의 거리 및 예상 도착시간을 파악할 수 있으며, 대리기사와의 통화 및 취소 기능도 제공한다.

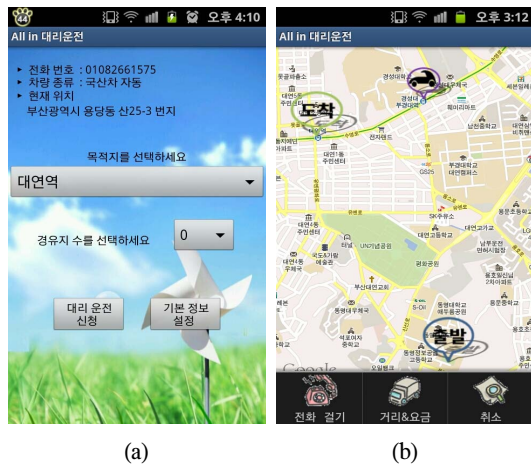


그림 2. 고객용 어플리케이션의 제공 기능
(a) 대리운전 신청 화면 (b) 부가 기능들
Fig. 2 The functions of user application
(a) requesting window (b) additional functions

3.3.2. 대리기사용 어플리케이션

대리기사가 기사용 어플리케이션을 통해 시스템에 로그인하면 위치 정보가 중계 서버에 전송되며 지도상에 그림 3(a)와 같이 표시된다. 이 때 대리기사는 “대기” 상태이며, 이전 위치로부터 변경 500m 이상 이동하면 위치 정보가 재전송된다.

대리기사가 콜센터의 대리운전 배정 메시지를 수신한 후 해당 메시지를 클릭하면 “수락/거절” 창이 그림 3(b)와 같이 나타난다. 지정된 시간 내에 “수락” 버튼을 누르면 “수락” 상태가 되며, 고객의 위치가 지도상에 표

시된다. 대리기사가 출발지에 도착한 후 “운행시작” 버튼을 누르면 목적지 정보와 운행 시작을 알리는 팝업창이 나타나며 대리기사는 “운행” 상태가 된다. 그리고 기사의 위치 정보는 목적지에 도착할 때까지 중계 서버에 주기적으로 전송되며, 목적지에 도착한 후 “운행종료” 버튼을 누르면 다시 “대기” 상태가 된다.

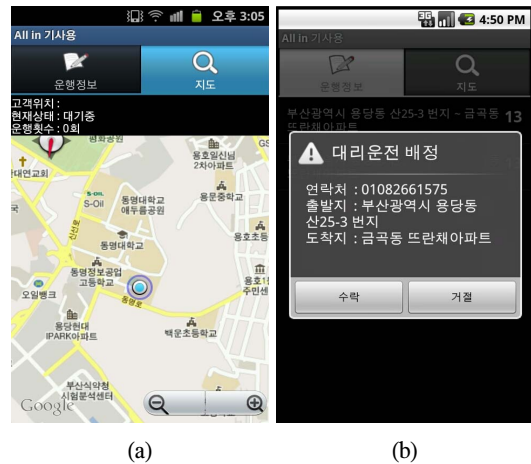


그림 3. 대리기사용 어플리케이션의 제공 기능
(a) 상태와 위치 표시 (b) 대리운전 배정 메시지
Fig. 3 The functions of a proxy driver application
(a) driver's state and location (b) assignment message

3.3.3. 콜센터와 중계 서버

콜센터는 대리기사들의 위치와 상태를 지속적으로 파악하며, 대리운전 신청을 접수한 후 고객으로부터 최단 거리에 위치한 대리기사를 선정하고 신청 처리 내역을 관리한다. 이를 위해 콜센터는 대리운전 접수 화면과 관리 화면으로 구성된다. 접수 화면은 그림 4와 같이 접수창과 지도, 그리고 신청의 처리 현황으로 구성되어 있다.

접수창에서 “대리기사 표시” 기능을 선택하면 대리기사들의 상태와 위치가 지도상에 표시된다. 대리기사의 상태에 따라 아이콘의 모양이 달라지며, 마우스 포인터를 아이콘 위에 올리면 대리기사의 식별자가 표시된다. 그림 5에서 대기 중인 대리기사의 식별자는 dr10001임을 알 수 있다.

콜센터의 신청 처리 과정은 다음과 같다. 1단계는 접수 과정이며, 신청 방식에 따라 다르게 처리된다. 상담원

이 고객과 통화할 때는 신청 내역을 접수 화면에 직접 입력한다. 그러나 온라인 주문에서는 접수부터 대리기사 선정까지 모든 과정이 자동으로 처리된다. 2단계는 요금 산출 과정이며, “조회” 버튼을 누르면 접수 내역에 근거하여 운행거리와 요금이 산정된다. 3단계는 기사 선정 과정이며, “서비스요청” 버튼을 누르면 접수 내역이 데이터베이스에 저장되고 대리기사를 선정한다.

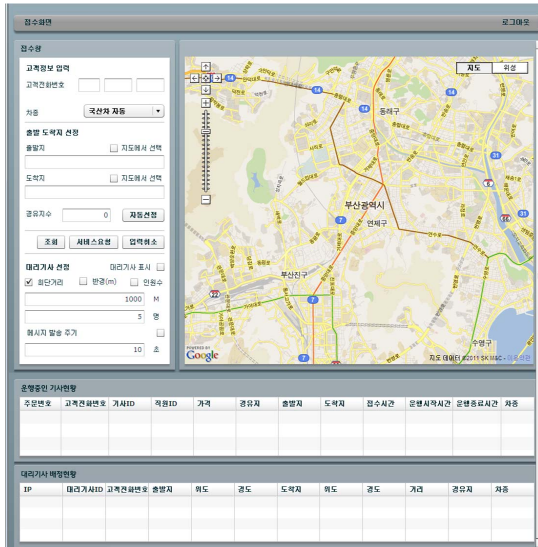


그림 4. 접수 화면
Fig. 4 The receipt window

대리기사 선정은 고객의 현재 위치로부터 최단거리에 위치한 대리기사를 최대 5명까지 후보로 선발한다. “반경”과 “인원수” 옵션의 추가적인 설정이 가능하며, 해당 옵션을 설정하면 지정된 반경 내의 대리기사들을 최단거리순으로 인원수만큼 선발한다. “메시지 발송 주기”를 설정하면 대리운전 배정 메시지를 1순위 후보자에게 우선적으로 발송하며 설정된 시간만큼 응답을 기다린다. 응답 메시지가 설정된 시간 내에 도착하지 않거나 “거절” 메시지가 도착하면 차순위 후보자에게 메시지가 재발송되며, 대리기사 선정시까지 이 과정이 반복된다. 콜센터가 “수락” 메시지를 수신하면 기사 선정이 완료되며 고객에게는 대리기사 정보를, 대리기사에게는 고객 정보를 발송한다. 4단계는 처리결과 표시 과정이며, 대리기사 선정 후 접수화면 하단에 대리운전 배정 내역이 표시된다.

한편 중계 서버는 고객용 어플리케이션과 대리기사용 어플리케이션, 콜센터 사이에서 메시지를 송수신하며, 콜센터의 지시에 따라 데이터베이스 서버에 대한 접근을 담당한다.

3.3.4. 데이터베이스 설계 및 데이터베이스 서버

본 논문에서는 5개의 테이블로 구성된 데이터베이스를 그림 6과 같이 설계하였다. 데이터베이스는 대리운전 신청과 처리에 관련된 정보들을 저장하며 중계 서버가 요청하는 정보를 제공한다.



그림 5. 대리기사들의 상태와 위치 표시
Fig. 5 The state and location of proxy drivers

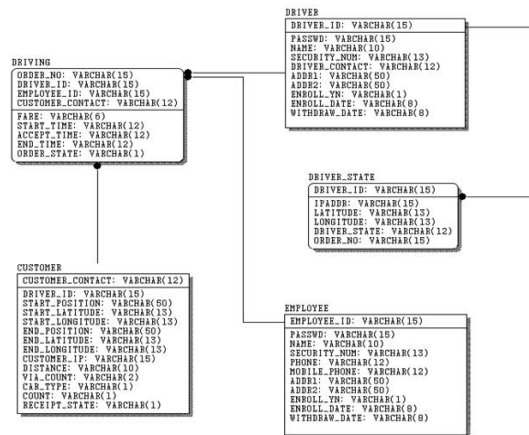


그림 6. 데이터베이스의 ER 다이어그램
Fig. 6 The ER Diagram of Database

- Driver 테이블: 대리기사들의 개인 정보 관리에 사용된다.
- Employee 테이블: 대리운전업체 직원들의 개인 정보 관리에 사용된다.
- Driver_state 테이블: 대리기사의 현재 위치와 상태 정보 관리에 사용된다.
- Driving 테이블: 콜센터에 접수된 대리운전 신청의 처리내역 관리에 사용된다.
- Customer 테이블: 대리운전 신청 및 처리 내역 관리에 사용된다.

3.4. 메시지 정의

대리운전 시스템의 구성 요소간 정보 송수신에 사용되는 메시지들의 종류는 표 1과 같다.

표 1. 메시지의 종류
Table. 1 The sorts of message

역할	메시지 형식	
대리기사-중계서버통신	로그인 요청	10#아이디#패스워드#IP어드레스#
	로그아웃 요청	11#아이디#
	로그인 성공	12#
	로그인 실패	13#실패원인
	로그아웃 성공	14#
	로그아웃 실패	15#실패원인
	대리운전 배정	30#아이디#위도#경도#
	배정 수락	31#아이디#
	배정 거절	32#아이디#
	운행 시작	33#아이디#
	운행 완료	34#아이디#
	운행 취소	35#
위치정보 송신	40#아이디#위도#경도#	
중계서버-콜센터통신	대리기사 위치정보 수신	50#
	대리기사 상태 수신	51#
	대리운전 배정	52#출발지#도착지#차종
	배정 수락	53#
배정 거절	54#	
고객-중계서버통신	대리운전 신청	60#전화번호#시작위치#위도#경도#도착지#위도#경도#경유지#차종
	대리운전 신청 취소	61#
	대리기사 정보 조회	62#
	대리기사 정보 저장	63#
	신청 성공	64#
	대리기사 위치 요청	65#고객전화번호
	대리기사 위치요청 성공	66#처리상태#대리기사위도#대리기사 경도
	대리기사 위치요청 실패	67#

3.5. 대리기사 상태와 신청처리 상태

본 논문에서는 대리기사와 신청 정보의 관리를 위해 대리기사와 신청처리 상태를 다음과 같이 정의하였다. 대리기사는 시스템에 로그인한 시점부터 로그아웃할 때까지 표 2의 상태들 중 하나를 갖는다. 신청 정보는 접수 시점부터 대리운전 종료시까지 표 3의 상태들 중 하나를 갖는다.

표 2. 대리기사의 상태
Table. 2 The state of a proxy driver

종류	코드	의미
대기	0	대리운전 배정을 기다림
수락	1	대리기사로 선정됨
운행	2	대리운전 중

표 3. 대리운전 신청의 처리 상태

Table. 3 The state of a proxy driving request

종류	코드	의미
접수	0	콜센터가 대리운전 신청을 접수
수락	1	대리기사 선정 완료
진행	2	출발지로 이동 및 대리운전 중
종료	3	목적지에 도착
취소	4	대리운전 신청이 취소됨

3.6. 기존 고객용 어플리케이션들과의 기능 비교

기존의 대리운전 신청용 어플리케이션들과 본 논문에서 구현한 고객용 어플리케이션의 기능 비교는 표 4와 같다.

표 4. 기존 어플리케이션들과 본 논문의 기능 비교
Table. 4 The comparison between user applications

항목	종류	코리아 드라이브	로지소프트	이루은	본 논문
기본정보설정		×	×	×	○
출발지/도착지 설정	위치 입력	위치입력/지도에서 선택	위치입력/지도에서 선택	위치입력/설정 정보에서 선택	
위치 표시	고객	고객	고객/대리기사	고객/대리기사	
대리운전신청	온라인 주문	전화걸기/온라인주문	온라인 주문	전화걸기/온라인주문	
업체 선정	지정	지정	선택	지정	
주행거리표시	×	○	×	○	
요금 표시	○	○	○	○	
경유지 설정	○	○	○	○	

코리아드라이브[12]는 대리운전 신청시 출발지와 도착지를 고객이 입력한다. 대리운전 신청 후 지도에 고객의 위치만 표시된다. 대리운전 신청은 온라인 주문 방식을 지원하며, 업체는 지정되어 있다. 요금 표시와 경유지 설정 기능을 제공한다.

로지소프트[13]는 대리운전 신청시 출발지와 도착지를 직접 입력하거나 지도에서 선택할 수 있다. 신청 후 지도에 고객의 위치만 표시되며, 전화걸기 방식과 온라인 주문 방식을 지원한다. 업체는 지정되어 있으며, 주행거리와 요금 표시, 경유지 설정 기능을 제공한다.

이루온[14]은 출발지와 도착지를 직접 입력하거나 지도에서 선택할 수 있다. 신청 후 고객과 대리기사의 위치가 지도에 모두 표시되며, 온라인 주문 방식을 지원한다. 업체를 선택할 수 있으며, 요금 정보와 경유지 설정 기능을 제공한다.

본 논문에서는 기본정보 설정 기능을 제공하므로 출발지와 도착지는 설정된 값 중에서 선택하거나 직접 입력할 수 있다. 신청 후 고객과 대리기사의 위치가 지도에 모두 표시된다. 신청은 전화걸기 방식과 온라인 주문 방식을 지원하며, 업체는 지정되어 있다. 주행거리와 요금 표시, 경유지 설정 기능을 제공한다.

IV. 대리운전 시스템의 성능 분석

본 장에서는 전화걸기 방식과 온라인 주문 방식의 신청 처리시간을 비교함으로써 구현된 대리운전 시스템의 성능을 분석한다.

구현된 시스템에서 신청 처리시간은 그림 7과 같이 구성되며, 성능 분석을 위한 파라미터(단계별 소요시간)들은 표 5와 같다. 콜센터의 메시지 발송 주기와 DB 서버 내부의 처리시간은 각각 20초와 0초로 가정한다.

기존의 대리운전 시스템과 유사한 박상성[1]은 CTI(Computer Telephony Integration)을 이용하여 신청 처리시간을 이전의 10~15분에서 7~10분으로 줄였는데, 이것은 신청에서 대리기사 도착까지의 시간이다. 그리고 기존 시스템에서는 대리운전 배정 메시지를 대리기사들에게 동보 전송하므로 경우에 따라서는 원거리의 기사가 배정될 수도 있지만 구현된 시스템은 최단거리의 기사에게 전송하므로 동일한 전화걸기 방식이라도 고객의 대기시간이 평균적으로 감소하게 된다.

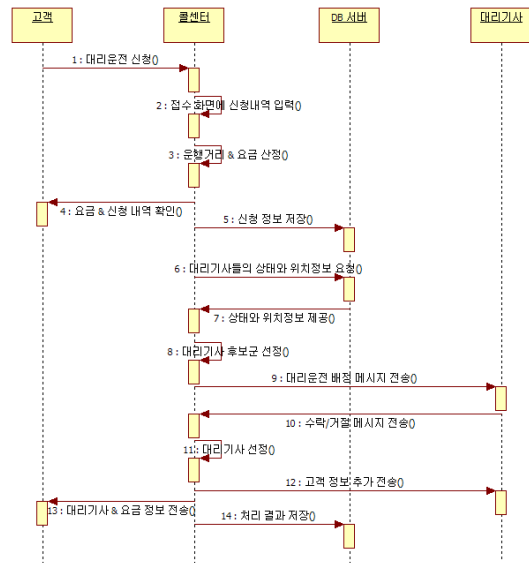


그림 7. 대리운전 신청 처리시간의 구성
Fig. 7 The composition of a call processing time

표 5. 시스템 파라미터
Table. 5 System parameters

종류	설정값(sec)	의미
$T_{1(direct)}$	60	콜센터 호출에 의한 대리운전 신청
$T_{1(online)}$	20	프로그램을 통한 온라인 신청
T_4	20	신청 내역 확인 및 요금 고지
$T_{8(direct)}$	3	대리기사 후보군의 수동 선정
T_9, T_{12}, T_{13}	5	콜센터의 SMS 전송지연 시간
T_{10}	25	대리기사의 응답 시간 (발송 주기+전송지연 시간)
$T_{11(direct)}$	3	대리기사 수동 선정의 소요 시간

전화걸기 방식의 처리시간(T_{direct})은 고객과 콜센터간 통화 및 수작업을 고려할 때 다음과 같이 산출되며, 결과값은 121초이다.

$$T_{direct} = T_{1(direct)} + T_4 + T_{8(direct)} + T_9 + T_{10} + T_{11(direct)} + T_{12}$$

온라인 주문 방식은 접수부터 기사 선정까지 모든 과정이 자동으로 처리되므로 해당 부분은 0초로 산정한다. 처리시간(T_{online})은 다음과 같이 산출되며, 결과값은 58초이다.

$$T_{online} = T_{1(online)} + T_9 + T_{10} + T_{12}$$

두 방식을 비교하면 온라인 주문 방식의 처리 시간이 전화걸기 방식에 비해 48%만큼 절감되며, 1시간동안 처리할 수 있는 신청건수는 전화걸기 방식이 29건, 온라인 주문 방식이 62건으로 평균 209% 정도 증가한다.

한편, 온라인 주문 비율(α)을 온라인주문 건수/총 신청건수로 정의할 때 대리운전 신청의 유효 처리시간 ($T_{effective}$)은 다음과 같다.

$$T_{effective} = (\alpha \times T_{online}) + ((1-\alpha) \times T_{direct})$$

대리운전 신청건수가 100개와 200개일 때 온라인 주문 비율을 0%, 25%, 50%, 75%, 100%의 5단계로 변경하면서 유효 처리시간을 비교한 결과는 그림 8과 같다. 즉, 온라인 주문 비율이 높을수록 처리시간이 감소하므로 콜센터 상담원의 업무량은 감소한다. 그리고 상담원이 통화중이면 추가 접수가 불가능한 전화걸기 방식에 비해 온라인 주문 방식에서는 즉시 접수가 가능하므로 고객의 대기시간을 단축시키는 효과도 제공한다.

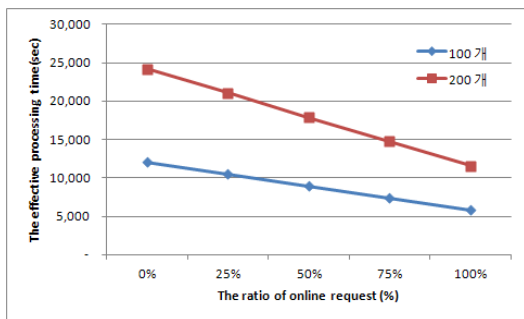


그림 8. 온라인 주문 비율에 따른 처리시간 비교
Fig. 8 The comparison of processing time depending on the ratio of online requests

V. 결 론

본 논문에서는 스마트폰을 활용하여 고객과 대리기사에게 기존의 대리운전 시스템보다 편리하고 양질의 서비스를 제공할 수 있는 3-tier 구조의 대리운전 시스템을

을 설계하고 구현하였다.

구현된 시스템은 대리기사의 위치 정보를 이용하여 배정된 기사의 현재 위치와 고객의 예상 대기시간을 산정할 수 있으므로 서비스에 대한 고객 만족도를 높일 수 있다. 콜센터는 대리기사들의 위치와 상태를 파악하고 있으므로 고객으로부터 최단 거리에 위치한 기사를 우선적으로 배정한다. 그리고 대리기사는 고객의 위치 정보가 어플리케이션의 지도에 표시되므로 고객의 위치로 정확하게 이동하여 신속한 서비스를 제공할 수 있으며, 서비스 품질 향상에 따른 수입 증대도 기대할 수 있다. 한편 3-tier 구조의 대리운전 시스템은 특정 시간대에 신청 건수가 급격히 증가하더라도 시스템에 과도한 부하를 유발하지 않는다. 또한 시스템 성능 분석을 통해 온라인 주문 비율이 높을수록 대리운전 접수와 처리의 자동화를 통해 콜센터 상담원의 업무량 감소와 대리운전 서비스 제공에 소요되는 대기시간 단축 효과도 제공함을 확인할 수 있었다. 따라서 업체 입장에서는 상담원의 증원을 최소화하면서 서비스 품질을 강화하여 고객 만족도를 향상시키고 비용 절감과 시장 점유율 확대 효과도 거둘 수 있다.

향후에는 고객의 서비스 활용 이력과 다양한 요구 사항에 근거하여 개인별 특성을 도출하여 개인별 맞춤형 서비스를 제공할 수 있는 시스템을 구축할 예정이다.

참고문헌

- [1] 박상성, 정원교, 신영근, 장동식, "CTI를 이용한 콜센터 시스템 개발: 대리운전 시스템," IE Interfaces, 제 20권, 제3호, pp. 309-314, 2007.
- [2] SK Telecom, "PDA기반 모바일 대리운전 서비스," Biz News Letter, 제15호, 2006.
- [3] 교통문화운전본부, "우리나라의 대리운전 현황", http://smartdriving.co.kr/main/main.p-hp?dir=smart_driving&mode=pd_info_03.
- [4] 박희근, 운전서비스시스템 및 운전서비스 제공방법, 공개특허 특2003-0076938, 2003.
- [5] 송익배, 황인혁, 자기위치 측정 및 지리정보 시스템을 이용한 택시와 대리운전 배차 시스템, 등록특허 10-0704409, 2007.

[6] 김성민, 최현숙, 위치정보를 이용한 대리기사 자동 연결 시스템, 등록특허 10-0789025, 2007.

[7] 주식회사 이루온, 위치기반의 대리운전 고객 접수/처리 장치, 등록특허 10-0953830, 2010.

[8] 연홍석, 생크 주식회사, 대리운전 서비스 시스템, 등록특허 10-1012501, 2011.

[9] 김갑호, 허송, 대리운전 호출시스템, 등록특허 10-0999778, 2010.

[10] Jeong-Gook Koh, "Proxy Driving Service System using a Location-based service," *Journal of Electronics and Computer Science*, Vol. 11, No. 2, pp. 59-64, 2009.

[11] Jeong-Gook Koh, "Design and Implementation of Android Applications for Proxy Driving Services," *Journal of Electronics and Computer Science*, Vol. 12, No. 1, pp. 43-46, 2010.

[12] 코리아드라이브, <http://www.koreadrive.com>

[13] 로지소프트, <http://www.logisoft.co.kr>.

[14] 이루온, <http://www.eluon.com>.

[15] Channu Kambalyal, "3-Tier Architecture," <http://channukambalyal.tripod.com/NTierArchitecture.pdf>.

[16] Scott Clinton, "Developing for Multi-tier Distributed Computing Architectures with Delphi Client/Server Suite 2.0," <http://dn.embarcadero.com/article/10343>.

[17] Rahman Mahmoodi, "3-tier architecture in C#," http://www.codeproject.com/KB/arc-hitecture/tree_tier_architecture.aspx.

[18] Mai-lan Tomsen, "Build Reliable and Scalable N-Tier Applications that Run on Both Windows NT and UNIX," *Microsoft Systems Journal*, pp. 20-27, 1998.

[19] 김정훈, 구글의 안드로이드 프로그래밍, 성안당, 2009.

[20] 리토 마이어, 프로페셔널 안드로이드 애플리케이션 개발, 제이펍, 2009.

[21] 옥상훈, 예제로 배우는 Adobe 플렉스, 에이콘출판사, 2008.

[22] 제프 태퍼, Adobe FLEX 3 실전 트레이닝 북, 위키북스, 2008.

[23] BlazeDS, <http://opensource.adobe.com/wiki/display/blazeds/BlazeDS>.

[24] MySQL, <http://www.mysql.com>.

저자소개



고정국(Jeong-Gook Koh)

1999년 부산대학교 컴퓨터공학과
공학박사

1999년~현재 동명대학교
컴퓨터공학과 부교수

※ 관심분야: 운영체제, 컴퓨터활용교육, 인터넷 응용