

형그리

배 고 플 때 가 장 먼 저 찾 는 플 랫 폼

맨

CONTENTS

01 제안플랫폼 소개

02 동작구동1 - DB 중심

03 동작구동2 - 시나리오 중심

04 동작 시현

05 구현 일정 및 역할 분담

01 제안플랫폼 소개

01 헝그리맨이란

헝그리맨이란

배고플 때 가장 먼저 찾는 플랫폼

HUNGRYMEN

Customer-Store-Rider을 한번에 연결하는 배달플랫폼이다.
다른 플랫폼과 다른점은 Store와 Rider 간의 광고 연결 서비스다.



STORE



CUSTOMER



RIDER

02 동작구동1

01 Store

02. Customer

03. Rider

04. Ordering

05. Advertishment

DB중심 : Store

Field	Type	NULL	Key	Default
_no	int(11)	NO	PRI	NULL
id	varchar(45)	NO		NULL
password	varchar(45)	NO		NULL
storename	varchar(45)	NO		
address	varchar(45)	NO		
category	varchar(45)	NO		

02 동작구동1

01. Store

02 Customer

03. Rider

04. Ordering

05. Advertisement

DB중심 : Customer

Field	Type	NULL	Key	Default
_no	int(11)	NO	PRI	NULL
id	varchar(45)	NO		NULL
password	varchar(45)	NO		NULL
address	varchar(45)	NO		NULL

02 동작구동1

01. Store

02. Customer

03 Rider

04. Ordering

05. Advertisement

DB중심 : Rider

Field	Type	NULL	Key	Default
_no	int(11)	NO	PRI	NULL
id	varchar(45)	NO		NULL
password	varchar(45)	NO		NULL
ridername	varchar(45)	NO		

02 동작구동1

01. Store

02. Customer

03. Rider

04 Ordering

05. Advertishment

DB중심 : Ordering

Field	Type	NULL	Default
_no	int(11)	NO	NULL
storename	varchar(45)	NO	NULL
menuname	varchar(128)	NO	NULL
ridername	varchar(45)	YES	NULL
storestatus	int(11)	YES	NULL
riderstatus	int(11)	YES	NULL
time_stamp	datetime	YES	CURRENT

02 동작구동1

01. Store

02. Customer

03. Rider

04. Ordering

05 Advertisement

DB중심 : Advertisement

Field	Type	NULL	Key	Default
_no	int(11)	NO	PRI	NULL
storename	varchar(45)	NO		NULL
ridername	varchar(45)	YES		NULL

03 동작구동2

01 Store

02. Customer

03. Rider

04. 플랫폼 요소

시나리오 : Store

1. 로그인 혹은 회원가입

- 로그인 : ID와 PW 동일 여부 확인 후
맞으면 로그인 성공 메시지
- 회원가입 : ID PW 동일 여부 확인 후
동일한 것이 없으면 회원가입 성공

2. 조리완료 선택

해당 Store와 관련된 Ordering DB의 List를
가져와서 원하는 List 번호를 선택하여
라이더가 매칭 됐는지 확인하고 조리 완료로 변경

3. 광고 등록

광고 등록 명령을 받으면 Advertisement Db에
상점 정보를 업데이트

4. 로그아웃

로그아웃하여 다시 1번을 수행할 수 있도록 함

03 동작구동2

01. Store

02 Customer

03. Rider

04. 플랫폼 요소

시나리오 : Customer

1. 로그인 혹은 회원가입

- 로그인 : ID와 PW 동일 여부 확인 후
맞으면 로그인 성공 메시지
- 회원가입 : ID PW 동일 여부 확인 후
동일한 것이 없으면 회원가입 성공

2. 카테고리 선택

Store 데이터베이스에 있는 카테고리
양식 / 중식 등을 출력해서 Client가 선택

3. 상점 선택 후 메뉴 선택

카테고리 내에 있는 Store들을 모두 출력하고
메뉴를 선택하게끔 함

4. 주문 정보 확인

주문 정보를 Client에게 전송하여 자신의
주문 내역을 확인할 수 있게끔 함

03 동작구동2

01. Store

02 . Customer

03 Rider

04. 플랫폼 요소

시나리오 : Rider

1. 로그인 혹은 회원가입

- 로그인 : ID와 PW 동일 여부 확인 후
맞으면 로그인 성공 메시지
- 회원가입 : ID PW 동일 여부 확인 후
동일한 것이 없으면 회원가입 성공

2. 주문 선택

Ordering 데이터 베이스를 통해 들어온 주문을
확인 후 배달을 원하는 주문이 있으면 상태 변경

3. 광고 등록

store가 올린 광고 중 원하는 광고를 선택하여
Advertisement DB 정보를 변경

4. 배달 완료

배달을 완료하면, Ordering DB에 있는
주문 상태를 변경

03 동작구동2

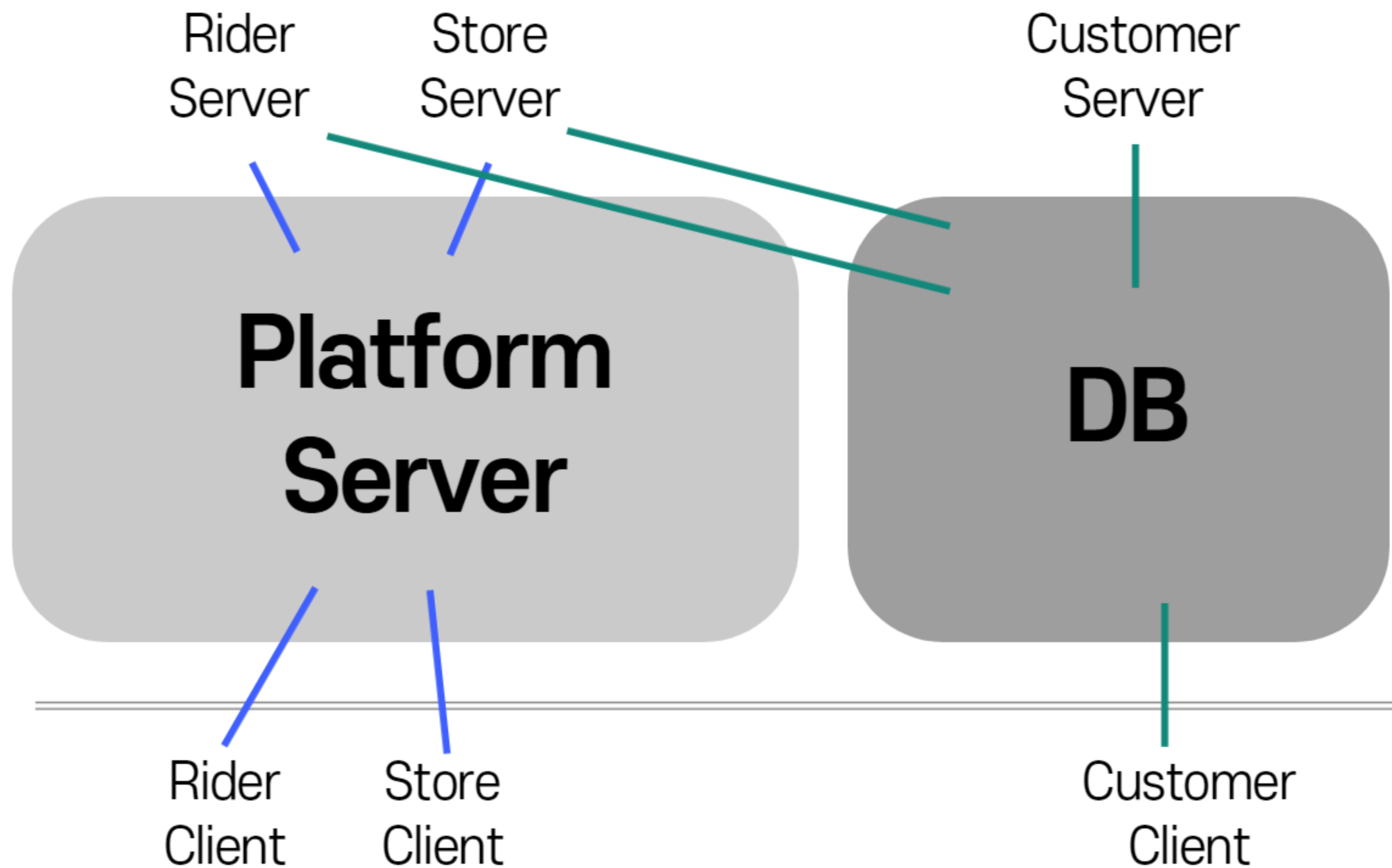
01. Store

02. Customer

03. Rider

04 플랫폼 요소

플랫폼 시스템 구조도



03 동작구동2

01. Store

02. Customer

03. Rider

04 플랫폼 요소

플랫폼 요소

```
/*
 * type      0은 client, 1은 server
 * number    0은 customer, 1은 store, 2는 rider
 */
typedef struct Destination {
    int type;
    int number;
    SOCKADDR_IN addr;
} Destination;

/*
 * 라우팅의 기본인 방향 구조체다.
 */
typedef struct PlatformProtocol {
    Destination start;
    Destination end;
    int flag;
    char data[950];
} PlatformProtocol;
```

[코드1] 플랫폼 주요 구조체

```
//서버서 서버로 (서버간 서비스 활용) (자기자신 시 플랫폼의 디비 접근 활용)
else if (platformbuf->end.type == 1 && platformbuf->start.type == 1) {
    SOCKADDR_IN destination;
    int wrongflag = 0;
    //자기자신의 루프인 경우 플랫폼의 데이터베이스 접근을 요구합니다.
    if (platformbuf->end.number == platformbuf->start.number) {
        MYSQL_RES* res;
        MYSQL_ROW row;

        if (mysql_query(conn, platformbuf->data) != 0) {
            printf("오류다.");
        }

        res = mysql_store_result(conn);
        memset(DataFrom, 0x00, sizeof(700));
        int j = 0;
        while (row = mysql_fetch_row(res)) {
            for (int i = 0; i < mysql_field_count(conn); i++) {
                strcpy(DataFrom[j][i], row[i]);
            }
        }
        strcpy(platformbuf->data, (char*)DataFrom);
        retval = sendto(sock, (char*)platformbuf, sizeof(PlatformProtocol), 0,
            (SOCKADDR*)&clientaddr, sizeof(SOCKADDR_IN));
        if (retval == SOCKET_ERROR) {
            err_display("sendto()");
            continue;
        }
    }
}
```

[코드2] 플랫폼 API 해당 코드

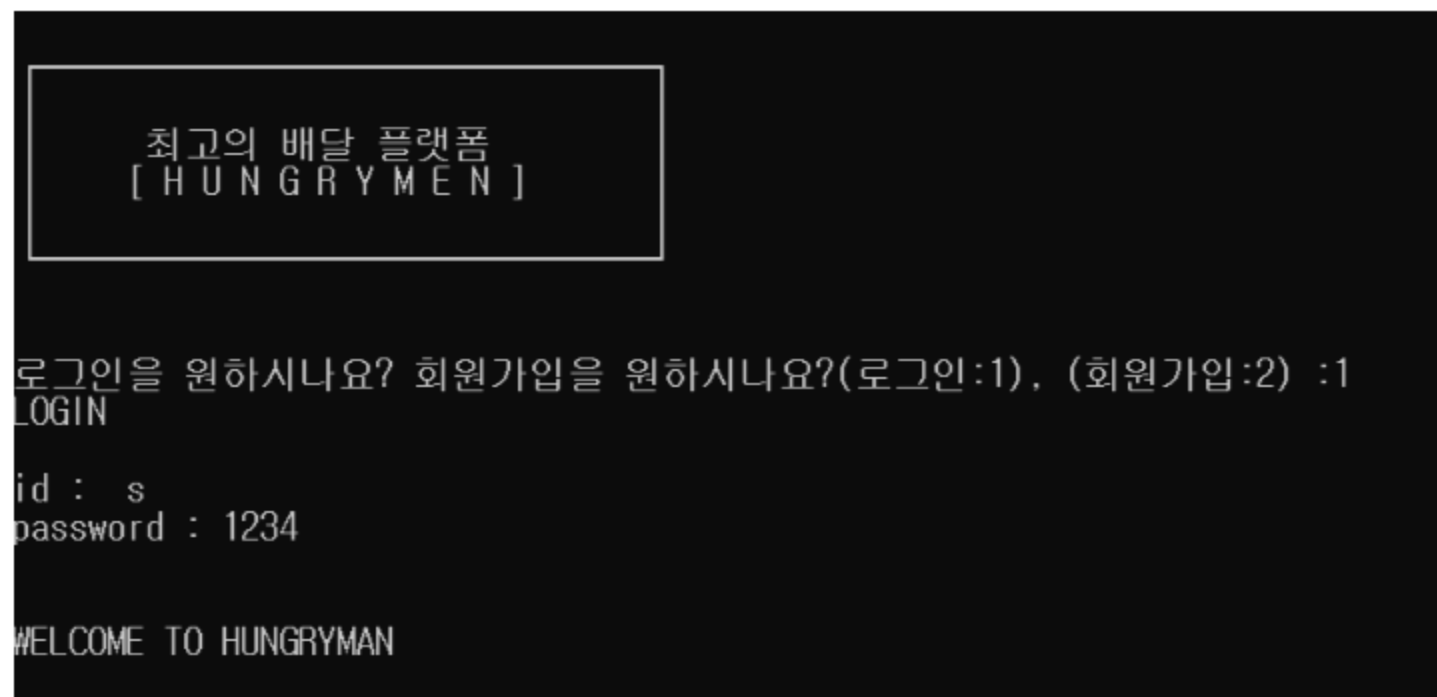
04 동작 시현

01 Store

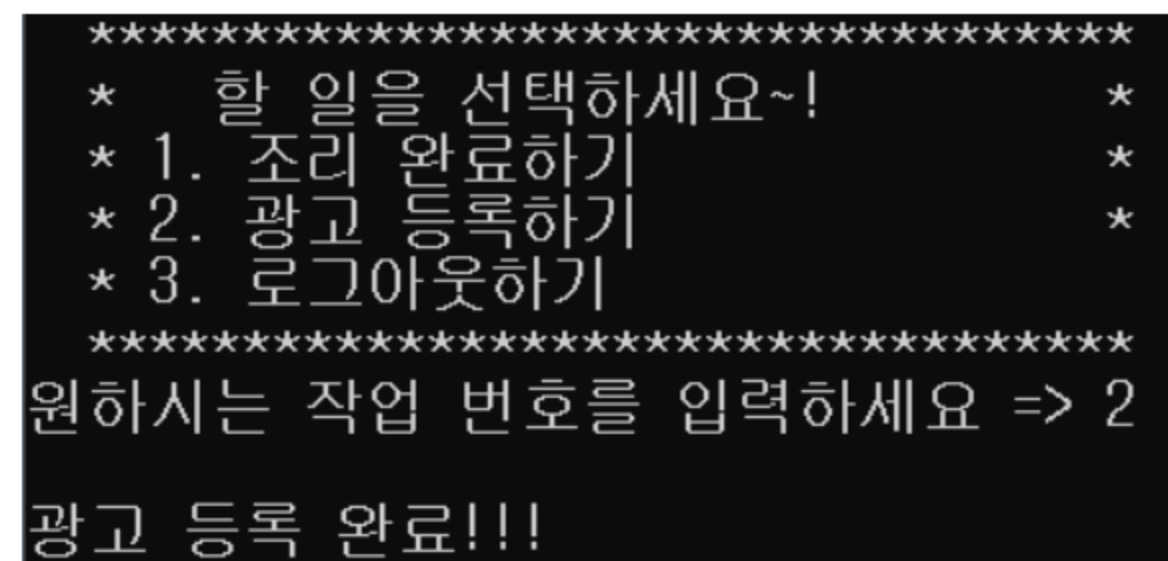
02. Customer

03. Rider

동작 시현 캡처



[캡처1] 로그인 및 회원가입



[캡처2] 광고 등록

04 동작 시현

01 Store

02. Customer

03. Rider

동작 시현 캡처

원하시는 작업 번호를 입력하세요 => 1

오더 list 1
18-06
짬뽕
경기도 수원시
조리 완료

오더 list 2
20-06
라면
경기도 안산시
조리 완료

오더 list 3
09-06
짜장면
경기도 수원시
조리 대기

오더 list 3
09-06
짜장면
경기도 수원시
조리 대기

라이더가 연결된 조리 완료할 메뉴를 리스트에서 선택해주세요 : 3

업데이트 완료!!!

* 할 일을 선택하세요~! *
* 1. 조리 완료하기 *
* 2. 광고 등록하기 *
* 3. 로그아웃하기 *

원하시는 작업 번호를 입력하세요 =>

[캡처3] 조리 완료로 상태 변경

04 동작 시현

01. Store

02 Customer

03. Rider

동작 시현 캡처

데이터베이스 insert된 모습

```
[HUNGRY MEN]

=====
      진행 할 작업을 선택하세요.
1. 로그인
2. 회원가입

=====
입력 : 1

ID 입력 : alswl222
PW 입력 : akqk

해당 유저가 없습니다.
=====
      진행 할 작업을 선택하세요.
1. 로그인
2. 회원가입

=====
입력 : 1

ID 입력 : alswlkku
PW 입력 : alswl1997

로그인 완료 !

===카테고리선택===
1. 한식
```

[캡처1] 로그인

```
1. 로그인
2. 회원가입

=====
입력 : 2

ID 입력 : alswlkku
PW 입력 : alswl1997

회원이 존재합니다.

      진행 할 작업을 선택하세요.
1. 로그인
2. 회원가입

=====
입력 : 2

ID 입력 : thewlwt
PW 입력 : elqlqlq

회원가입 완료

===카테고리선택===
1. 한식
2. 양식
3. 중식
```

```
Welcome to the MySQL monitor.  Copyright (c) 2000, 2017, Oracle and/or its affiliates.  Other names may be trademarks of their respective owners.

Type 'help;' or 'h;' for help. Type 'quit;' or 'q;' to quit the MySQL monitor.

mysql> use netsof
Database changed
mysql> select *from customer;
+----+-----+-----+
| _no | id      | password |
+----+-----+-----+
| 4   | alswlkku | alswl1997 |
| 5   | tjddnjs19 | ldlw19 |
| 6   | hello   | adsad |
| 7   | qweqew | qwrjqjr |
| 8   | weqew   | wekqwelj |
| 9   | alswl19912 | kdjw |
| 10  | thewlwt | elqlqlq |
+----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

[캡처2] 회원가입

04 동작 시현

01. Store

02 Customer

03. Rider

동작 시현 캡처

```
6 | eosong | su-si | (null) | 0 | 0 | 2020-06-29 08:07:13 | suwon-si wooman-dong 32-3
+-----+-----+-----+-----+-----+-----+-----+-----+
Microsoft Visual Studio 디버그 콘솔
4. 일식
원하는 카테고리 번호 : 2
1) eosong
2) ajou-matjip
상점 이름 : eosong
메뉴 이름 : su-si
주문지 주소 : suwon-si wooman-dong 32-3

----- 주문접수완료 -----
상점 이름 : eosong
메뉴 이름 : su-si
주소 : suwon-si wooman-dong 32-3
```

[캡처3] 주문내역 확인

04 동작 시현

01. Store

02. Customer

03 Rider

동작 시현 캡처

```
*****
*   진행 할 작업을 선택하세요   *
*   1. 로그인                   *
*   2. 회원 가입                 *
*****
```

원하시는 작업 번호를 입력하세요 => 1

```
-----
아이디를 입력하세요 => gugu
비밀 번호를 입력하세요 => 1234
-----
```

로그인 되었습니다. 다음 기능으로 넘어갑니다.

```
*****
*   할 일을 선택하세요~!       *
*   1. 주문 접수하기          *
*   2. 광고 등록하기          *
*   3. 배달 완료하기          *
*   4. 종료하기               *
*****
```

원하시는 작업 번호를 입력하세요 => 1

```
----- 선택 리스트 -----
1 => 가게 이름 : ㅎㅎ      1 => 손님 주소 : 수원시      1 => 가게 주소 : ㅇㅇㅇ      1 => 날짜 : 08-06
!! 되돌아가기 => 0번을 입력하세요
-----
```

[캡처1] 로그인 및 주문 접수

04 동작 시현

01. Store

02. Customer

03 Rider

동작 시현 캡처



[캡처2] 광고등록

04 동작 시현

01. Store

02. Customer

03 Rider

동작 시현 캡처

```
*****
*   할 일을 선택하세요~!   *
* 1. 주문 접수하기       *
* 2. 광고 등록하기       *
* 3. 배달 완료하기       *
* 4. 종료하기            *
*****
원하시는 작업 번호를 입력하세요 => 3

----- 선택 리스트 -----
1 => 가게 이름 : 구들보쌈      1 => 손님 주소 : 수원      1 => 가게 주소 : 수원시      1 => 날짜 : 09-06
!! 되돌아가기 => 0번을 입력하세요

----- 원하는 리스트 번호를 선택하세요 -----
1
정상적으로 처리되었습니다.

*****
*   할 일을 선택하세요~!   *
* 1. 주문 접수하기       *
* 2. 광고 등록하기       *
* 3. 배달 완료하기       *
* 4. 종료하기            *
*****
원하시는 작업 번호를 입력하세요 =>
```

[캡처3] 배달 완료 상태 변경

05 구현 일정 및 역할 분배

01 구현 일정 및 역할 분배

상세 구현 계획

- 1. Core 플랫폼 서버 설계 및 구현
- 2. Customer Client 설계 및 구현

박민지

- 1. Core 플랫폼 서버 설계 및 구현
- 2. Store Client 설계 및 구현

김수영

- 1. 제안 플랫폼 설계 및 구현
- 2. Rider Client 설계 및 구현

구교현

6월 4일 - 6일
: 피드백 수정 및 계획 수립

6월 8일 - 14일
기본 설계 및 기본 오류 수정

6월 15일 - 16일
1차 테스트 및 디버깅

6월 18일 - 6월 28일
상세 설계 및 최종 디버깅

6월 29일
최종 구현

감사합니다
