<뚝배기>

<MrGreat>

설계 산출물

|  |  |
| --- | --- |
| 문서버전 | 1.1 |
| 문서 ID | SE-2020-002 |
| 최종변경일 | 2020-11-05 |
| 문서상태 | 알파 |

요 약

미스터 대박 디너 서비스(Mr.Great)의 설계 산출물을 기술.

서브 시스템의 구성과 각 서브 시스템의 구조를 기술.

주요 산출물

* 아키텍처도
* 클래스도
* 교류도
* 상태도

표 1 문서 변경 기록

|  |  |  |  |
| --- | --- | --- | --- |
| 문서이름 | <뚝배기> <MrGreat> 설계 산출물 | | |
| 문서 ID | SE-2020-002 | | |
| 버전 | | 변경일 | 설명 |
| 1 | 0 | 2020-11-4 | 아키텍처도를 추가했다.  “Staff Client” 클래스도를 추가했다.  “Server” 클래스도를 추가했다. |
| 1 | 2020-11-4 | “Customer Client” 클래스도를 추가했다.  “Customer Client” 서브시스템 세부설계 설명을 추가했다.  “Staff Client” 서브시스템 세부설계 설명을 추가했다.  “Server” 서브시스템 세부설계 설명을 추가했다. |

1 개 요

1.1 목 적

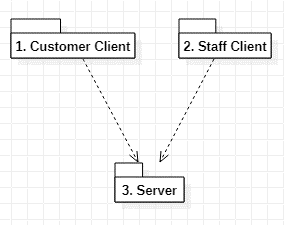
본 문서는 미스터 대박 디너 서비스(Mr.Great)의 산출물을 기술한다.

* 전체 시스템이 어떤 서브 시스템으로 구성되어 있는지 “아키텍처도”를 통해 보여준다.
* 서브 시스템이 어떤 클래스로 구성되어 있고 클래스들간에 어떤 관계가 있는지를 “클래스도”를 통해서 보여준다.
* 객체간에 발생하는 동적인 행위를 “교류도”를 통해서 보여준다.
* 한 객체의 상태 변화를 “상태도”를 통해서 보여준다.

1.2 참고 문헌

없음.

2 아키텍처도



전체 시스템은 아래의 서브 시스템으로 구성된다.

- Customer Client 시스템

- Staff Client 시스템

- Sever 시스템

**고객 Client 시스템**

- 서버와 통신하며 고객에게 정보를 보여주는 역할을 한다.

- Javascript를 이용하여 작성된다.

**직원 Client 시스템**

- 서버와 통신하며 직원에게 정보를 보여주는 역할을 한다.

- Javascript를 이용하여 작성된다.

**Server 시스템**

- 메뉴, 고객 정보, 주문 정보를 담고 있으며 주문 저장 및 알림의 역할을 한다.

- Node.js를 이용하여 작성된다.

3 서브시스템 세부 설계

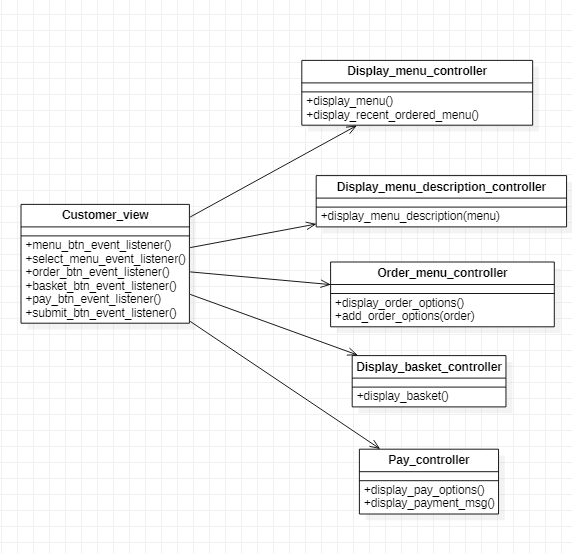
3.1 “Customer Client” 서브시스템 세부 설계

“Customer Client" 시스템은 고객에게 필요한 정보를 ”Server" 시스템으로부터 받아와서 보여준다. 또한, 고객으로부터 입력을 받고 "Server" 시스템으로 전송한다.

“Customer Client” 시스템의 주요 함수는 다음과 같다.

* **각종 display**
  + "Customer Client" 시스템에서 event\_listener가 호출되었을 때 필요한 정보를 보여준다.
* **각종 event\_listener**
  + 각종 버튼이 눌렸을 때 호출되는 함수로 "Server" 시스템에 특정 요청을 한다.

- “Customer Client” 서브시스템 메인 클래스도



**Customer\_view**

* “Customer Client" 시스템은 Customer\_view 클래스를 통해 사용자의 입력을 처리한다.
* Customer\_view 클래스는 입력에 따라 Controller 클래스의 함수를 호출한다.

**각종 Controller**

* “Customer Client" 시스템은 각종 Controller 클래스를 통하여 “Server"에 요청을 보낸다.

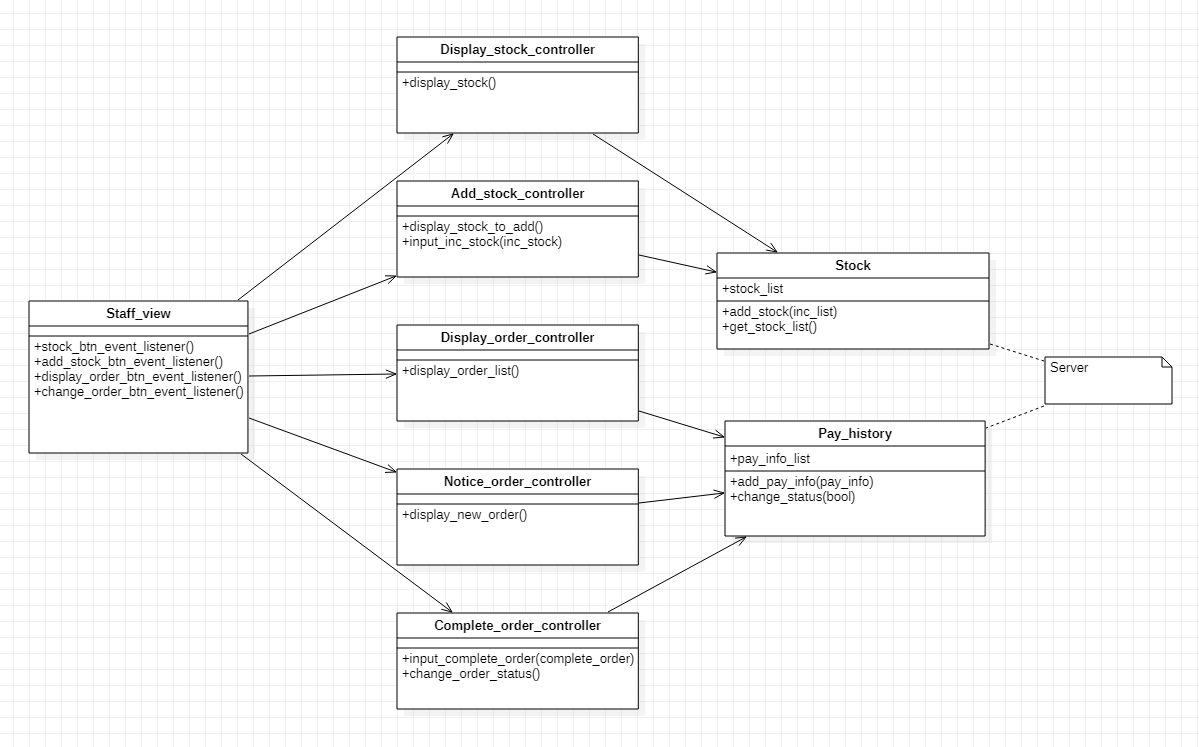
3.2 “Staff Client” 서브시스템 세부 설계

“Staff Client” 시스템은 주문 내역과 재고 내역을 “Server” 시스템으로부터 받아와서 보여준다. 그리고 납품으로 늘어난 재고의 양과 완료된 주문 목록을 입력으로 받고 “Server” 시스템으로 전송한다.

“Staff Client” 시스템의 주요 함수는 다음과 같다.

* **각종 display**
  + "Staff Client" 시스템에서 event\_listener가 호출되었을 때 필요한 정보를 보여준다.
* **각종 event\_listener**
  + 각종 버튼이 눌렸을 때 호출되는 함수로 "Server" 시스템에 특정 요청을 한다.

- “Staff Client” 서브시스템 메인 클래스도



**Staff\_view**

* “Staff Client" 시스템은 Staff\_view 클래스를 통해 사용자의 입력을 처리한다.
* Staff\_view 클래스는 입력에 따라 Controller 클래스의 함수를 호출한다.

**각종 Controller**

* “Staff Client" 시스템은 각종 Controller 클래스를 통하여 “Server"에 요청을 보낸다.

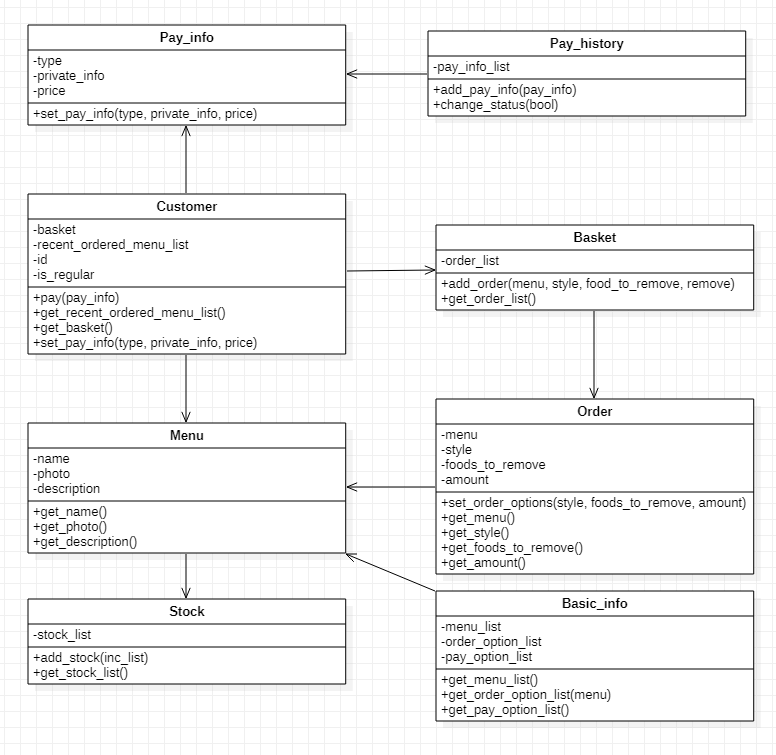
3.3 “Server” 서브시스템 세부 설계

“Server" 시스템은 고객에게 메뉴 관련 정보를 제공하고 주문 정보를 저장한다. 또한, "Customer Client”, “Staff Client”에게 받은 요청을 처리하는 역할을 한다.

“Server" 시스템의 주요 함수는 다음과 같다.

* **set\_order\_options**
  + “Customer Client"에서 주문 option을 전송받고 주문 정보를 생성한다.
* **add\_order**
  + “Customer Client” 시스템으로부터 전송받은 정보를 토대로 주문 정보를 생성하고 장바구니에 추가한다.
* **set\_pay\_info**
  + “Customer Client” 시스템에서 결제 정보를 받는다.
* **add\_pay\_info**
  + “Customer Client” 시스템에서 결제가 성사된 pay\_info를 저장한다.
* **change\_status**
  + “Staff Client”에서 전송받은 주문을 완료 상태로 전환한다.
* **add\_stock**
  + “Staff Client”에서 전송받은 납품 정보를 반영한다.
* **pay**
  + 결제 시스템에게 결제 정보를 넘기며 결제를 요청한다.

- “Server” 서브시스템 메인 클래스도



**Basic\_info**

* 미스터 대박 서비스의 기본적인 정보를 담고 있는 클래스이다.
* “Server” 시스템은 Basic\_info를 통해 “Customer Client” 시스템에게 메뉴 정보, 메뉴에 따른 주문 옵션 정보, 결제 정보를 넘겨준다.

**Menu**

* 메뉴 정보를 저장하는 type을 나타내는 클래스이다.
* 예시 사진과 메뉴 설명에 대한 정보를 담고 있다.

**Customer**

* 고객을 나타내는 클래스이다.
* 고객을 식별하는 id, 단골 고객 여부, 장바구니, 최근 주문 내역을 저장하고 있다.
* Pay 함수를 통해 장바구니에 담긴 주문에 대해 최종 결제를 할 수 있다.

**Basket**

* 장바구니 type 클래스이다.
* 장바구니는 주문 정보의 집합이며 add\_order를 통해 장바구니에 주문 정보를 추가할 수 있다.

**Order**

* 주문 정보를 저장하는 type을 나타내는 클래스이다.
* 메뉴, 스타일, 뺄 음식, 수량 정보를 저장하고 있으며 set\_order\_options로 주문 정보를 설정할 수 있다.

**Pay\_info**

* 결제 정보를 저장하는 type을 나타내는 클래스이다.
* 결제 수단, 결제 관련 개인 정보, 최종 결제 가격 정보를 저장하고 있으며 set\_pay\_info로 결제 정보를 설정할 수 있다.

**Stock**

* 재고의 정보를 저장하는 type을 나타내는 클래스이다.
* 재고의 종류, 재고의 수량을 저장하고 있으며 add\_stock을 통해서 납품된 재고를 추가할 수 있다.

**Pay\_history**

* 성사된 주문의 정보를 저장하는 클래스이다.
* 새로 성사된 주문을 add\_pay\_info를 통해서 추가할 수 있다.
* change\_status를 통해서 배달이 완료된 주문을 처리 완료 상태로 전환할 수 있다.

4. 교류도

4.1

5. Transaction 객체 상태도