
Software Requirements Specification (SRS)

전북대학교

소프트웨어공학

버전 1.0 (2023.10.03)

김용현(201818716), 유지호(202111581), 이재연(202112043)

목차

목차	1
1. 소개	2
1.1 목적	2
1.2 프로젝트 범위	2
1.3 용어사전	2
1.4 참고자료	2
1.5 문서개요	2
2. 전체 설명	3
2.1 시스템 환경	3
2.2 기능적 요구사항 및 사양	3
2.2.1 객실 예약 및 관리	3
2.2.2 체크인 및 체크아웃 관리	4
2.2.3 고객 서비스 및 편의시설	4
2.2.4 요금 및 결제 관리	5
2.2.5 리뷰 및 피드백	6
2.3 사용자 클래스와 특성	6
2.4 사용자 설명서	6
3. 시스템 요구사항	7
3.1 기술 스택	7
3.1.1 에디터	7
3.1.2 버전 관리	7
3.1.3 클라이언트(프론트엔드), 서버(백엔드) 공통	7
3.1.4 클라이언트(프론트엔드)	8
3.1.5 서버(백엔드)	8
3.2 클라이언트(프론트엔드)	9
3.2.1 비기능적 요구사항	9
3.2.2 기능적 요구사항	9
3.3 서버(백엔드)	11
3.3.1 비기능적 요구사항	11
3.3.2 기능적 요구사항	11
3.4 외부 인터페이스(결제 게이트웨이)	11

1. 소개

1.1 목적

이 호텔 예약 및 관리 시스템은 OO호텔에서 예약, 객실 관리, 고객 서비스 및 운영을 효율적으로 관리하기 위한 소프트웨어입니다. 이는 호텔 관리자, 직원, 그리고 고객 모두에게 이점을 제공하는 목적으로 설계되었습니다. 시스템의 목적과 특징, 인터페이스에 대해 설명하고 활동 주제마다 시스템의 수행 작업 등의 내용을 설명합니다. 또한 시스템이 외부 자극에 대해 어떻게 반응하는지, 어떠한 제약사항이 있는지 작성되었습니다.

1.2 프로젝트 범위

이 소프트웨어 시스템은 고객의 서비스 및 운영을 편리하게 하기 위한 웹 사이트입니다. 시스템은 고객이 수동으로 수행해야만 했던 예약 가능 여부와 서비스 정보 프로세스를 자동화하는 도구를 제공함으로써 고객의 편의를 극대화하도록 설계될 것입니다. 또한 해당 시스템은 직원과 호텔 관리자의 업무 효율성과 생산성을 극대화함으로써 이들의 요구를 충족시키고 동시에 고객의 사용이 용이할 것입니다. 세부적으로, 이 시스템은 직원이 고객의 사용 정보를 전산 처리하고 결제 정보를 기록할 수 있도록 설계되었습니다. 웹 서비스를 통해 고객은 호텔 내부 정보와 날짜에 따른 객실 유무를 살펴볼 수 있도록 하였고 호텔 관리자는 고객의 예약 승인 여부를 관리할 수 있습니다.

이 소프트웨어는 고객과 호텔 관리자 간의 의사소통을 용이하게 합니다. 호텔을 사용하고 난 후 고객은 시설에 대한 피드백을 웹 사이트를 통해 호텔 관리자에게 제공할 수 있습니다. 이러한 정보들과 결제 정보를 기록할 수 있는 데이터베이스가 이 시스템에 포함되어 있습니다.

1.3 용어사전

용어	정의
고객	웹 서비스를 이용하는 사람으로, 웹 사이트에 방문하여 원하는 정보를 얻은 후 그 서비스를 사용하고 피드백을 제공합니다.
직원	고객의 체크인 및 체크아웃을 전산 처리하고 결제 정보를 기록하는 사람입니다.
호텔 관리자	고객의 예약 승인 · 거부를 하는 사람으로, 웹 사이트에 객실 상태를 업데이트할 수 있으며 고객의 피드백을 수용합니다.
데이터베이스	데이터베이스(DB)는 이 시스템에서 모니터링하는 모든 정보를 수집합니다.

1.4 참고자료

① Software Engineering 10E Ian Sommerville

1.5 문서개요

2장에서는 제품의 기능에 대한 개요를 설명합니다. 해당 섹션은 프로젝트의 이해를 돕기 위한 비공식적인 요구사항을 설명하고, 3장에서의 기술적 요구사항 사양에 대한 컨텍스트를 구축하는 데 사용됩니다. 3장은 주로 개발자를 위해 작성된 섹션으로 제품의 기능에 대한 세부 사항들을 기술적인 용어로 설명합니다. 문서의 두 섹션 모두 동일한 소프트웨어 제품을 전체적으로 설명합니다. 그러나, 서로 다른 청중을 대상으로 하므로 서로 다른 언어를 사용합니다.

2. 전체 설명

2.1 시스템 환경

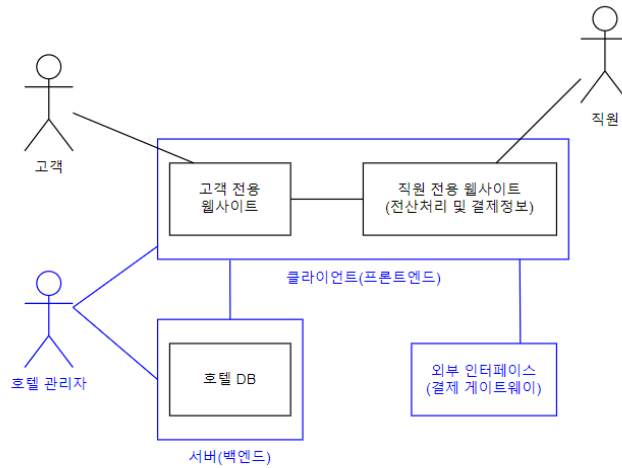


그림 - 시스템 환경

00호텔 예약 및 관리 시스템은 총 3명의 활동 주체와 클라이언트 및 서버 그리고 한 개의 외부 인터페이스가 있습니다. 온라인 고객들은 인터넷을 통해 고객 전용 웹사이트 혹은 직원 전용 웹사이트에 액세스 합니다. 호텔 관리자는 웹 사이트를 편집 및 업데이트합니다. 호텔 직원은 고객의 체크인과 체크아웃 서비스를 전산 처리 하고 결제 정보 등을 기록합니다.

2.2 기능적 요구사항 및 사양

이 섹션에서는 주요 기능들을 활동 주체들의 사용 사례를 통해 개별적으로 설명합니다.

2.2.1 객실 예약 및 관리

사용 사례1 : (고객) 웹 서비스 접근 후 객실 예약

사용 사례2 : (호텔 관리자) 웹 서비스 접근 후 객실 관리

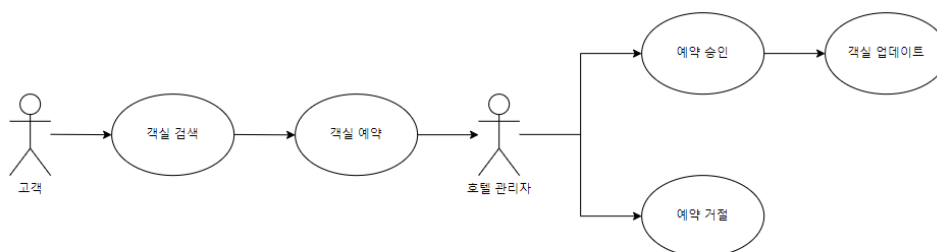


그림- 객실 예약 및 관리

간단한 설명

고객은 호텔의 객실을 검색하고 예약합니다. 호텔 관리자는 예약을 승인하거나 거절하며 객실 상태를 업데이트합니다.

단계별 설명

1. 고객은 웹 사이트에 접근하여 원하는 객실을 검색합니다.
2. 해당 객실을 예약합니다.
3. 호텔 관리자는 고객의 예약을 승인하거나 거절합니다.
4. 상황에 따라 객실의 상태를 웹 사이트에 업데이트합니다.

2.2.2 체크인 및 체크아웃 관리

사용 사례 : 체크인/체크아웃 후 웹 사이트 관리

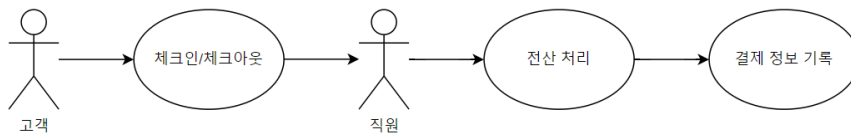


그림- 체크인 및 체크아웃 관리

간단한 설명

직원은 고객의 체크인 혹은 체크아웃을 전산 처리하고 결제 정보를 기록합니다.

단계별 설명

1. 고객은 예약한 객실을 체크인하거나 체크아웃합니다.
2. 직원은 외부 인터페이스를 호출하여 해당 객실에 대한 정보를 전산 처리하고 객실에 대한 결제 정보를 기록합니다.

2.2.3 고객 서비스 및 편의시설

사용 사례 : 호텔 내 서비스 검색 및 예약

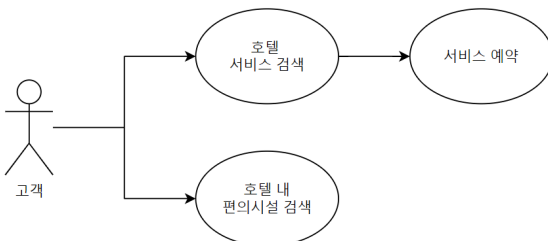


그림- 고객 서비스 및 편의시설

간단한 설명

고객은 호텔 내 편의시설과 서비스를 검색하고 예약 가능한 서비스를 예약합니다.

단계별 설명

1. 고객은 웹 사이트에 접근하여 호텔 내 서비스나 편의시설을 검색합니다.
2. 예약 가능한 서비스 중 원하는 서비스를 예약합니다.

2.2.4 요금 및 결제 관리

사용 사례 : 객실 예약 승인 후 결제

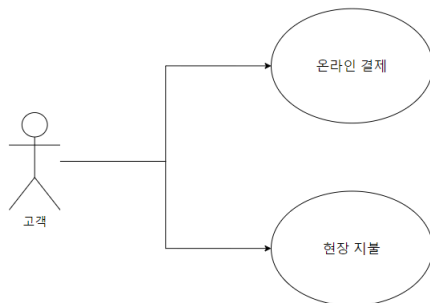


그림-요금 및 결제 관리

간단한 설명

고객은 자신이 예약한 객실에 대한 금액을 지불하기 위해 결제 방법을 선택합니다.

단계별 설명

이 단계에서 고객은 이미 예약 승인이 완료된 후 결제 단계로 넘어온 상태입니다.

1. 고객은 온라인 결제 또는 현장 지불을 선택합니다.
2. 온라인 결제 시 외부 서비스를 이용해서 정의되어 있는 인터페이스를 호출하여 결제를 완료합니다. 현장 지불 시 예약 당일 OO호텔에서 금액을 지불합니다.

2.2.5 리뷰 및 피드백

사용 사례 : 고객의 피드백 작성

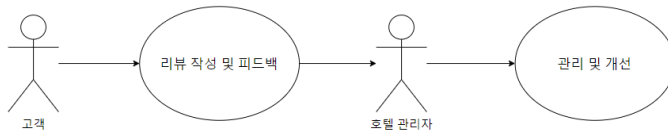


그림- 리뷰 및 피드백

간단한 설명

고객은 호텔의 객실 사용 및 서비스를 이용한 후 선택적으로 리뷰를 작성하여 호텔에 대한 피드백을 제공합니다.

단계별 설명

1. 고객은 온라인 저널에 액세스하여 리뷰 작성을 선택합니다.
2. 장점 및 단점을 기술하여 호텔에 대한 피드백을 제공합니다.
3. 호텔 관리자는 고객의 피드백을 볼 수 있으며 이를 관리하고 호텔을 개선합니다.

2.3 사용자 클래스와 특성

고객	검색 엔진과 같은 웹서비스를 사용할 수 있어야 합니다. (OS : 윈도우 10 이상 권장)
직원 / 호텔 관리자	웹서비스를 사용하는 것뿐만 아니라 웹서비스 관리 등 인터넷에 능숙해야 합니다. (OS : 윈도우 10 이상 권장)

2.4 사용자 설명서

고객	1. 객실 검색 및 예약 시 - 웹사이트 접속 > 객실 검색 > 원하는 객실 예약 요청 > 요청 승인 시 객실비용 결제 2. 편의시설 및 서비스 예약 시 - 웹사이트 접속 > 서비스 및 편의시설 검색 > 서비스 예약 3. 체크인 및 체크아웃 시 - 웹사이트 접속 > 객실 체크인/체크아웃 4. 리뷰 작성 시 - 웹사이트 접속 > 리뷰페이지 > 리뷰 작성
직원	1. 체크인 및 체크아웃 시 - 웹사이트 접속 > 객실 체크인/체크아웃 전산처리 > 결제정보 기록
호텔 관리자	1. 객실 예약 요청 시 - 웹사이트 접속 > 객실 관리페이지 > 승인 여부 결정 > 승인 시 객실 상태 업데이트 2. 리뷰 작성 시 - 웹사이트 접속 > 리뷰페이지 > 리뷰 관리 및 개선

3. 시스템 요구사항

해당 파트는 개발자 혹은 관리자를 위해 작성된 섹션이므로 전문용어를 사용합니다.

이는 '2.1 시스템 환경' 및 '2.2 기능적 요구사항 및 사양'에 대한 자세한 내용을 포함합니다.

3.1 기술 스택

3.1.1 에디터

① VScode

- VScode는 Microsoft가 개발한 소스 코드 편집기입니다. 프로그램이 매우 가벼우며 다양한 확장 기능을 통해 웹개발에 있어 높은 효율을 보입니다. 현재 시장점유율의 70%를 차지하고 있습니다.

3.1.2 버전 관리

① Github

- Github는 루비 온 레일즈로 작성된 분산 버전 관리 툴입니다. 팀원들 간의 협업 및 버전 관리에 있어 사용됩니다.

3.1.3 클라이언트(프론트엔드), 서버(백엔드) 공통

① Node.js / NPM

- Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 Javascript 런타임입니다. Javascript 프로그래밍 언어 기반의 클라이언트와 서버 시스템을 구현하기 위해 사용됩니다.
- NPM은 Node Package Manager의 약자로, 자바스크립트를 위한 패키지 관리자입니다. Node.js에서 패키지 관리를 위해 사용됩니다.

② ESLint / Prettier

- Javascript 언어를 통한 협업 시, 일관된 코드 설계 및 작성을 위해 사용됩니다.
- ESLint는 자바스크립트 코드에서 발견되는 문제시되는 패턴들을 식별하기 위한 정적 코드 분석 도구입니다.
- Prettier는 코드 포맷터(Code Formatter)의 한 종류로, 개발자가 작성한 코드를 정해진 코드 스타일에 따르도록 자동으로 변환해주는 도구를 말합니다.

3.1.4 클라이언트(프론트엔드)

다양한 기술들이 사용됩니다.

① HTML

- 웹 사이트의 모습을 기술하기 위한 마크업 언어입니다. 문서의 구조나 서식을 나타내기 위해 사용됩니다.

② CSS / SCSS / Tailwind CSS

- CSS는 HTML 등의 마크업 언어로 작성된 문서가 실제로 웹사이트에 표현되는 방법을 정해주는 스타일 시트입니다. 웹페이지의 정적 디자인을 할 때 주로 사용됩니다.
- SCSS는 CSS 전처리기로, 코드의 재사용과 가독성을 올려주고 다양한 기능들을 제공하여 CSS의 단점을 보완합니다. 효율적인 CSS개발을 가능하게 해줍니다.
- Tailwind CSS는 오픈 소스 CSS 프레임워크입니다. 효율적인 CSS개발을 가능하게 해줍니다.

③ Javascript / Typescript / React.js

- Javascript는 웹 브라우저 내에서 주로 사용되는 객체 기반의 스크립트 프로그래밍 언어입니다. 클라이언트(프론트엔드) 개발의 프로그래밍 언어로서 사용됩니다.
- Typescript는 Javascript의 단점을 보완하기 위해 만들어진 정적 타입 언어입니다. Javascript의 단점을 보완하여 효율적이고 안정적인 개발을 진행하기 위해 사용됩니다.
- React.js는 Node.js에서 동작하는 자바스크립트 라이브러리입니다. 리렌더링이 잦은 동적인 모던 웹에서 빠른 퍼포먼스를 내기 위해 사용됩니다.

④ Web Assembly

- Web Assembly는 웹 브라우저에서 실행되는 언어이자 바이트코드입니다. C/C++등의 언어의 사용을 통해 메모리와 CPU의 효율성을 높이고 최종적으로는 성능을 향상시키기 위해 사용됩니다.

⑤ Webpack

- Webpack은 모듈 번들러입니다. 브라우저에서 사용할 수 있도록 Javascript 파일을 번들링 혹은 패키징 해줍니다. 웹 어플리케이션의 빠른 로딩 속도와 높은 성능을 위하여 사용됩니다. (React.js에 기본적으로 탑재되어 있습니다.)

3.1.5 서버(백엔드)

다양한 기술들이 사용됩니다.

① Linux

- Linux는 리눅스 커널에 기반을 둔 오픈 소스 유닉스 계열 운영체제입니다. AWS등의 클라우드 서비스를 이용하여 서버를 구축할 때 기본적인 운영체제로 사용됩니다.

② Docker

- Docker는 리눅스의 응용 프로그램들을 프로세스 격리 기술들을 사용해 컨테이너로 실행하고 관리하는 오픈소스 프로젝트입니다. 코드를 더 빨리 전달하고, 운영을 표준화하고, 코드를 원활하게 이동하고, 비용을 절감할 목적으로 사용됩니다.

③ Express.js

- Express.js는 Node.js에서 웹 어플리케이션 혹은 API 서버를 구축하는데 가장 많이 사용되는 대표적인 프레임워크입니다. (이는 Java의 Spring, Python의 Django와 동일합니다.) 클라이언트(프론트엔드)와 서버(백엔드)간의 통신을 위해 사용됩니다.

④ MySQL

- MySQL은 오픈소스 관계형 데이터베이스 관리 시스템입니다. 고객 · 직원 · 호텔관리자 및 객실 · 결제정보 · 리뷰 · 편의시설 등의 서비스를 관계형 데이터베이스화 시켜 관리하기 위해 사용됩니다.

3.2 클라이언트(프론트엔드)

3.2.1 비기능적 요구사항

1. 기술스택

1-1. 호환성 오류 및 지원 중단등의 이슈가 발생하지 않는 이상, 3.1번의 기술 스택을 이용하여, 훗날 호텔 관리자들이 시스템 유지·보수를 가능하게 해야합니다.

2. 사용성

2-1. Chrome Browser, Firefox, Internet Explorer 11 이상을 필수적으로 지원해야 합니다.

2-2. 모바일·태블릿·PC등 다양한 기기에서 정상적으로 작동할 수 있도록 크로스 브라우징을 지원해야 합니다.

2-3. 시스템 사용자들이 버그 혹은 건의사항을 제보할 수 있도록하는 페이지가 제공되어야 합니다.

3. 성능

3-1. 알고리즘적인 부분은 Web Assembly로 작성하여 시스템 속도를 향상시켜야 합니다.

3-2. 네트워크 속도가 10Mbps 이상일 때, 모든 웹페이지 로딩 시간은 1초를 초과해서는 안됩니다.

3-3. 네트워크 속도가 10Mbps 이상일 때, 객실·서비스·편의시설 검색 시간은 버튼을 클릭한 이후 0.3초를 초과하면 안됩니다.

3-4. 웹페이지 성능 측정 방법 중 하나인 Google Pagespeed Insights에서, 웹과 모바일 모두 점수가 80점 이상을 넘겨야 합니다.

3.2.2 기능적 요구사항

1. 객실 예약 및 관리

외부 참조	섹션 2.2.1 객실 예약 및 관리
트리거	고객 전용 웹사이트의 '객실 예약 및 관리' 페이지에서 모든 필수 입력사항을 입력 후 '확인' 버튼을 클릭하면 실행됩니다.
방향성	클라이언트 -> 서버
전송되는 정보	사용자 번호(String), 객실 번호(String), 날짜(Time) 의 3가지 정보를 json 형태로 서버에 전달합니다.
서버에서의 처리	해당 정보를 가공하여 관계형 데이터베이스에 반영합니다.
참고	<p>* 사용자 관련 : 이름(String), 나이(Integer), 성별(Bool), 핸드폰번호(String) 등 사용자의 추가적인 정보는 회원가입 시 이미 서버에 저장되었으므로, 추가적인 데이터로 보낼 필요가 없습니다. (데이터 정규화가 잘 진행된 경우, 불필요한 데이터를 추가적으로 보내는 것은 성능의 감소로 이어집니다.)</p> <p>* 객실 관련 : 객실 타입(String), 객실 사이즈(Integer), 공실(입실) 여부(Bool) 등의 데이터 역시 이미 서버에 저장되었으므로, 추가적인 데이터로 보낼 필요가 없습니다.</p>

2. 체크인 및 체크아웃 관리

외부 참조	섹션 2.2.2 체크인 및 체크아웃 관리
트리거	직원 전용 웹사이트의 '체크인 및 체크아웃 관리' 페이지에서 입실 여부를 선택한 후 '확인' 버튼을 클릭하면 실행됩니다.
방향성	클라이언트 -> 서버
전송되는 정보	객실 번호(String), 공실(입실) 여부(Bool) 의 2가지 정보를 json 형태로 서버에 전달합니다.
서버에서의 처리	해당 정보를 가공하여 관계형 데이터베이스에 반영합니다.

3. 고객 서비스 및 편의 시설

외부 참조	섹션 2.2.3 고객 서비스 및 편의시설
트리거	고객 전용 웹사이트의 '고객 서비스 및 편의시설' 페이지에서 이용하고자 하는 서비스와 시간을 선택한 후 '확인' 버튼을 클릭하면 실행됩니다.
방향성	클라이언트 -> 서버
전송되는 정보	사용자 번호(String), 서비스 or 편의시설(String), 시간(Time)의 3가지 정보를 json 형태로 서버에 전달합니다.
서버에서의 처리	해당 정보를 가공하여 관계형 데이터베이스에 반영합니다.

4. 요금 및 결제 관리

외부 참조	섹션 2.2.4 요금 결제 및 관리
트리거	고객 전용 웹사이트의 '요금 결제 및 관리' 페이지에서 '결제' 버튼을 클릭하면 실행됩니다.
방향성	클라이언트 -> 외부 인터페이스
전송되는 정보	결제 및 카드 정보
인터페이스의 처리	결제 정보들이 외부 인터페이스(결제 게이트웨이)를 통해 전달된 후 결제가 완료됩니다.

5. 리뷰 및 피드백

외부 참조	섹션 2.2.5 리뷰 및 피드백
트리거	고객 전용 웹사이트의 '리뷰 및 피드백' 페이지에서 '확인' 버튼을 클릭하면 실행됩니다.
방향성	클라이언트 -> 서버
전송되는 정보	사용자 번호(String), 객실 번호(String), 별점(Integer), 내용(String)의 4가지 정보를 json 형태로 서버에 전달합니다.
서버에서의 처리	해당 정보를 가공하여 관계형 데이터베이스에 반영합니다.

3.3 서버(백엔드)

3.3.1 비기능적 요구사항

1. 기술스택

1-1. 호환성 오류 및 지원 중단등의 이슈가 발생하지 않는 이상, 3.1번의 기술 스택을 이용하여, 훗날 호텔 관리자들이 시스템 유지·보수를 가능하게 해야합니다.

1-2. 회사의 자체적인 서버 운용이 어려우므로, AWS등의 클라우드 서비스를 필수적으로 이용해야 합니다. 또한, 서버와 클라이언트간의 통신 최적화를 통해 한달 서버 운용비용을 100만원 아래로 유지해야 합니다. 서버는 필수적으로 Ubuntu Linux를 이용해야 합니다.

2. 성능

2-1. 최소 100명의 요청을 동시에 처리할 수 있는 규모의 서버를 운영하여야 합니다. 또한 Scalability 측면에서 훗날 확장이 용이해야 합니다. 100명을 초과하는 요청에 대해서는 대기 시간이 발생해도 괜찮습니다.

2-2. 24시간 내내 동일한 성능으로 서버 운용이 가능해야 합니다.

3. 데이터베이스

3-1. 데이터베이스는 필수적으로 무료로 제공되는 관계형 데이터베이스를 이용해야 합니다. Oracle 등의 추가적인 비용이 발생하는 데이터베이스를 사용해서는 안됩니다. MySQL이 아닌 다른 종류의 오픈소스 데이터베이스는 사용 가능합니다.

3-2. 관계형 데이터베이스는 필수적으로 정규화를 통해 중복되는 데이터를 최소화해야 합니다.

3.3.2 기능적 요구사항

1. '3.2 클라이언트 - 3.2.2 기능적 요구사항'과 동일합니다.

3.4 외부 인터페이스(결제 게이트웨이)

1. 결제 게이트웨이는 외부서비스를 통해 미리 정의되어있는 API를 이용하는 방식으로 개발되어야 합니다.
2. 카드결제, 휴대폰 결제, 네이버페이, 카카오페이, 토스, PAYCO 의 6개 결제 방식이 필수적으로 제공되어야 합니다.