애플리케이션 구현

* 애플리케이션 구현 개발환경 구축

|  |
| --- |
| 1. 프로젝트 주제 : MAFIA GAME 2. 개발 언어 : JAVA – NETWORK, THREAD, SWING 3. 개발 인원 : 5명 4. 개발 기간 : 1개월 ( 5/21일 ~ 6/22일) 5. 개발 H/W 사양  * 프로세서 : Intel® Core(TM) I7-6700 CPU @ 3.40GHz 3.41GHz * 메모리 : 16GB * OS : Windows10 64비트 운영체제 |

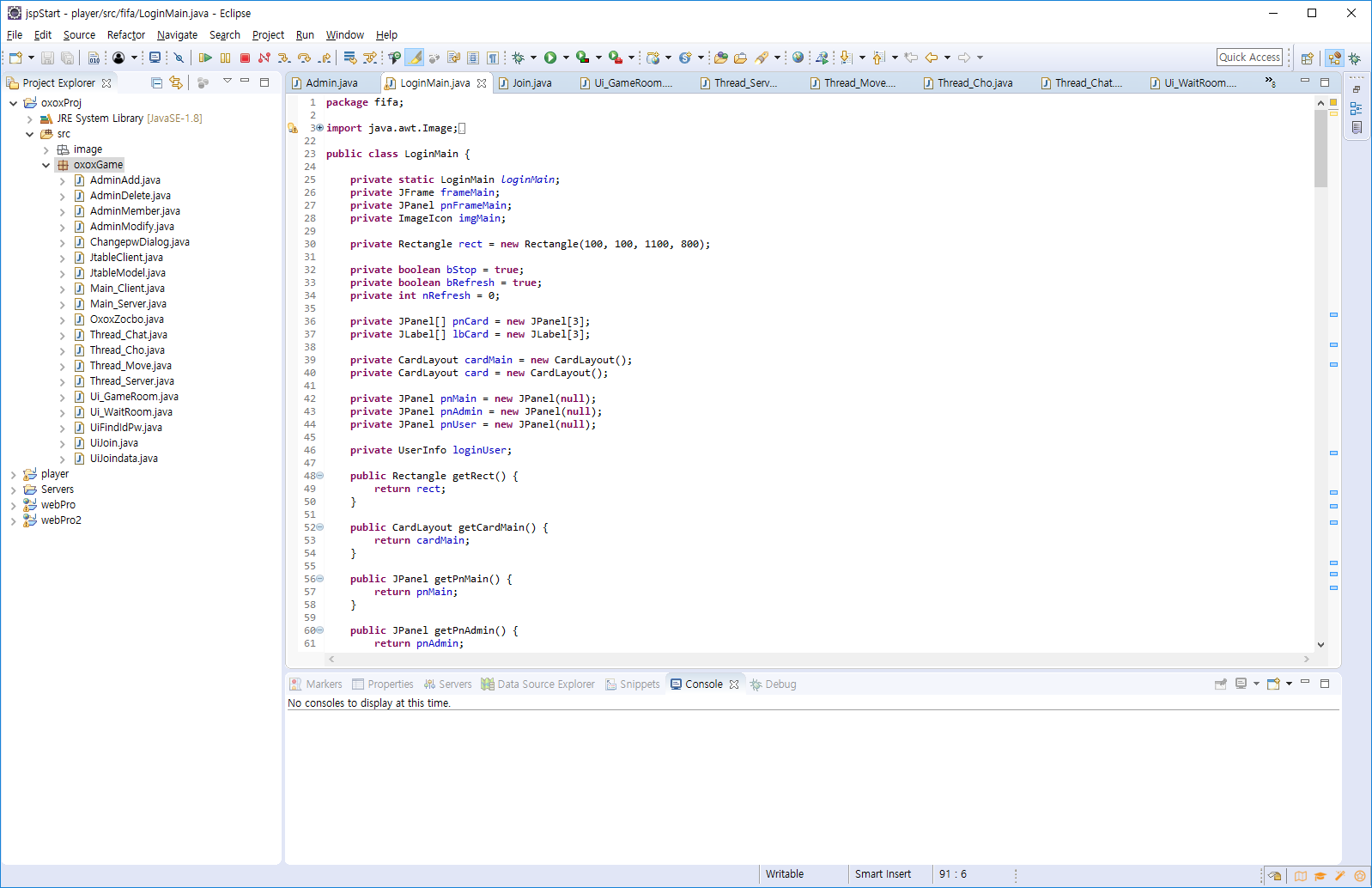
* 이클립스 설치



* 이클립스 실행



* 프로젝트 실행



공통 모듈 구현하기

* 공통 모듈 명세서 작성

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 업무기능 분류 | 테이블 조회 | | | |
| 프로그램ID | MAFIAGAME | | 프로그램명 | MAFIAGAME |
| 프로그램언어 | JAVA | | 프로그램 목록번호 | - |
| 프로그램 개요 | 참가자의 이미지를 클릭하여 투표를 받아 OUT 대상자 처리. | | | |
| 기능 구성도 | |  |  |  | | --- | --- | --- | | INPUT | PROCESS | OUTPUT | | 1. 클라이언트의 투표 버튼 클릭 | 1. 서버에서 투표 결과를 취합하여 최다 득표자를 OUT처리 | 1.최다 득표자의 클라이언트에 OUT 처리 및 서버를 통해 각 사용자에게 OUT 사실 통보 | | | | |
| I/O | Table ID | Table 명 | | 비 고 |
| I  O | JLabel  jLabel | jl1  jl2 | | -  - |

공통 모듈 구현하기

* 단위테스트 수행계획서

|  |  |  |  |
| --- | --- | --- | --- |
| 테스트케이스 ID | 0001 | 테스트케이스명 | 접속 및 시작 |
| 수행시스템 | NCS | 하위시스템 | - |
| 작성자 | 이도협 | 작성일자 | 2018/06/19 |
| 수행자 | 이도협 | 수행일자 | 2018/06/19 |

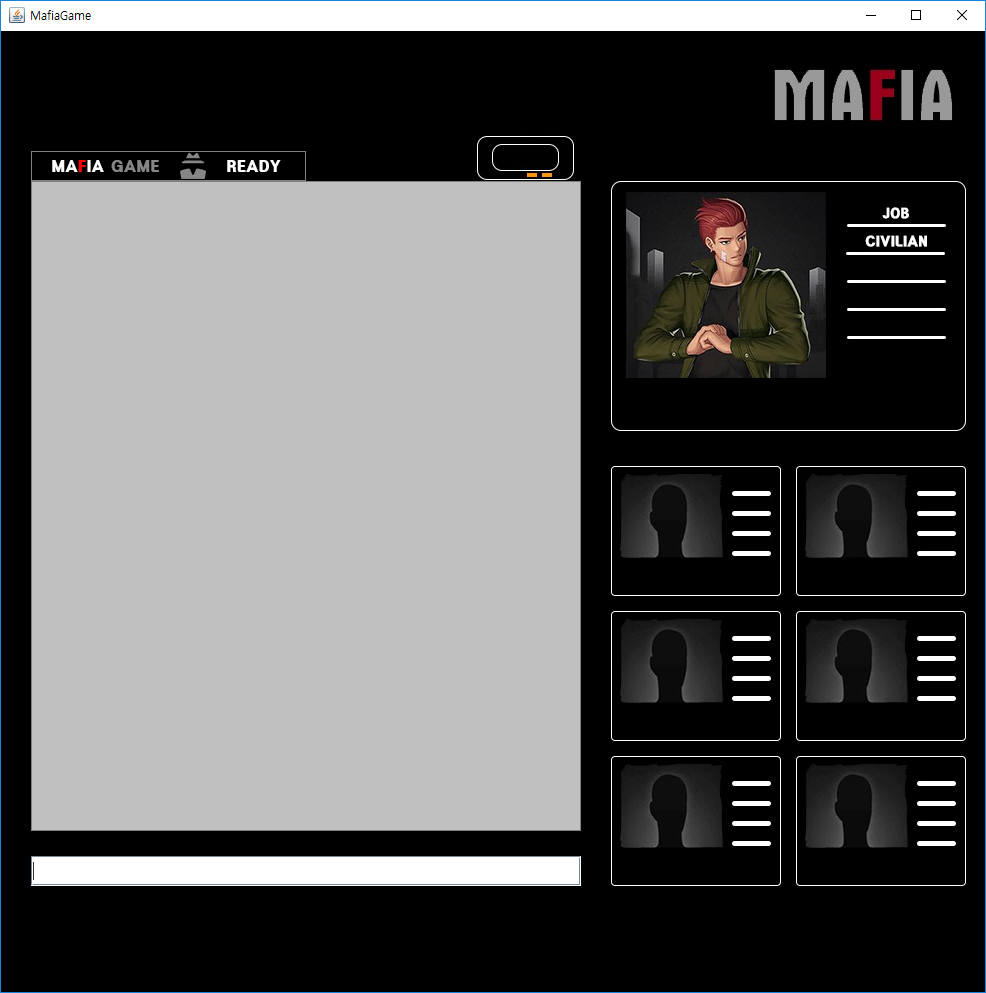
|  |  |
| --- | --- |
| 선행조건 | 서버가 동작중이고 IP를 확인하여 클라이언트를 실행시킨다 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Action | 예상응답 | 결과 | 비교 |
| 1 | YOUR NAME 클릭 | 내용이 사라지고 이름을 입력 할 수 있는 상태로 변경 | O |  |
| 2 | 이름 입력 | 이름이 2~12자로 입력하도록 제약조건 설정 | O |  |
| 3 | START 버튼 클릭 | 메인 페이지로 이동 | O |  |

* Login 실행



* 로그인 후 페이지 이동 확인



서버 프로그램 구현

* 서버 프로그램 테스트케이스

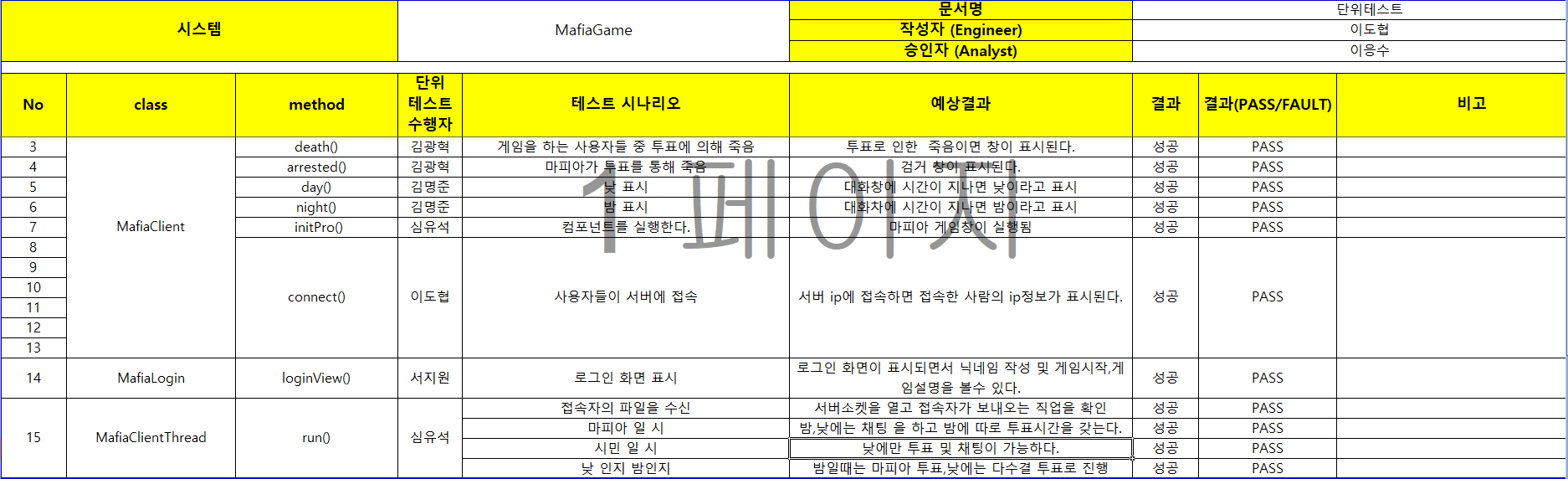
|  |  |  |  |
| --- | --- | --- | --- |
| 테스트케이스 ID | 0002 | 테스트케이스명 | 서버프로그램 구현 |
| 수행시스템 | NCS | 하위시스템 | 서버실행 |
| 작성자 | 이도협 | 작성일자 | 2018/06/19 |
| 수행자 | 이도협 | 수행일자 | 2018/06/19 |

|  |  |
| --- | --- |
| 선행조건 | NETWORK 연결을 준비하여야 한다. |

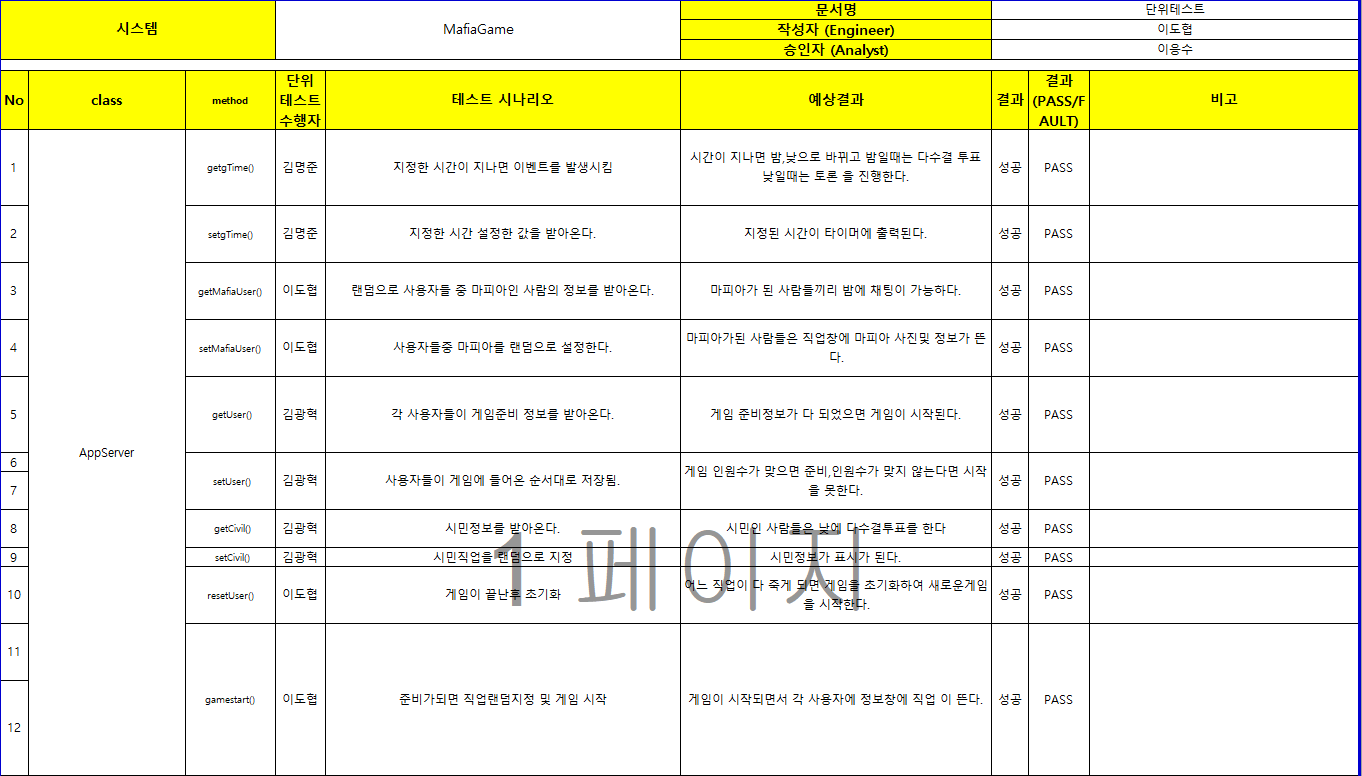
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Action | 예상응답 | 결과 | 비교 |
| 1 | IP/PORT 설정 | 연결할 IP, PORT를 설정한다. | O |  |
| 2 | 후행확인 |  | O |  |
| 3 |  |  |  |  |

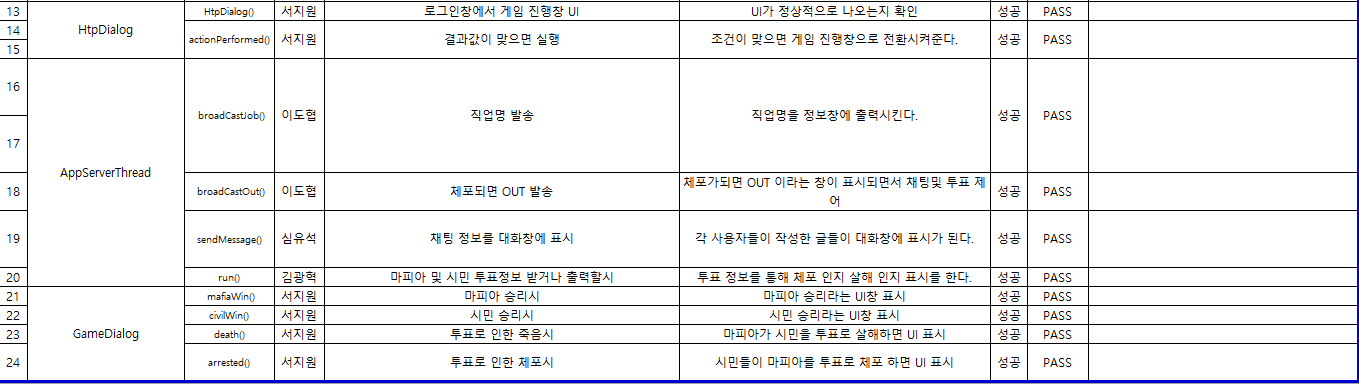
|  |  |
| --- | --- |
| 후행확인 | 설정한 IP/PORT 번호로 서버소켓을 생성한 후 Client를 대기한다. |

단위테스트 사용자



단위테스트 관리자





단위테스트 결함관리

* 결함관리 작업흐름도

(수정 시 첨부파일 참고)

화면 구현

유스케이스(Use Case)상세 예시

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U001 | UseCase명 | | 마피아게임 |
| 작성자 | 이도협 | 수정자 | | 이도협 |
| 최초 작성일 | 2018/06/19 | 최종 수정일 | | 2018/06/19 |
| Actor | 사용자 | | | |
| 사전 조건 | 사용자는 해당 게임에 접속하기 위해 서버의 IP를 확인 및 Client에 입력한다. | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액터 행위 | | 시스템 행위 | |
|  | 1.YOUR NAME 칸 클릭  3.이름을 입력한다.  5.START 버튼을 클릭한다..  7.상단에 READY 버튼을 클릭한다. | | 2.클릭 이벤트 처리를 통해 기존에 입력되어있던 ‘YOUR NAME’ 제거  4.제약조건 검사를 통해 2~12자로 입력되었는지 확인  6.서버에 접속하여 참가자+1  8.레디상태를 확인하고 참가자들이 전원 레디 버튼을 눌렀는지 확인 | |
| 대체 흐름 | 액터 행위 | | 시스템 행위 | |
|  |  | | 8-1.레디 인원 수가 참가자 전원의 숫자와 일치하지 않으면 대기  8-2.레디 인원 수=6 인 경우 게임을 시작 | |
| 예외 흐름 | 액터 행위 | | 시스템 행위 | |
|  |  | |  | |
| 포함/확장 |  | | | |
| 우선 순위 | 상 | | | |
| 사용 빈도 | 상 | | | |
| 업무 규칙 |  | | | |
| 특별요구사항 |  | | | |
| 비고 | 메인 페이지로 넘어온 순간 자신의 이름을 사용하여 채팅 | | | |

유즈케이스 다이어그램은 사용자의 관점으로 작성하고, SW 지식이  
깊지 않은 사용자라도 이해해야 하기 때문에 그 형식과 내용이 매우 간결하게 표현됩니다.

액티비티 다이어그램(Activity Diagram)예시

예

게임 시작 투표 시작

참가 인원이 6명인가

참가자 대기

레디

채팅입력

예

메인 화면 표시

만족하지 않음.

접속 이름 조건 확인

START 버튼 클릭

기존의 value 값 삭제

이름 입력

이름 입력창 활성화

이름 입력 선택

시스템

사용자

제약 조건을 모두 만족 하는가?

아니요

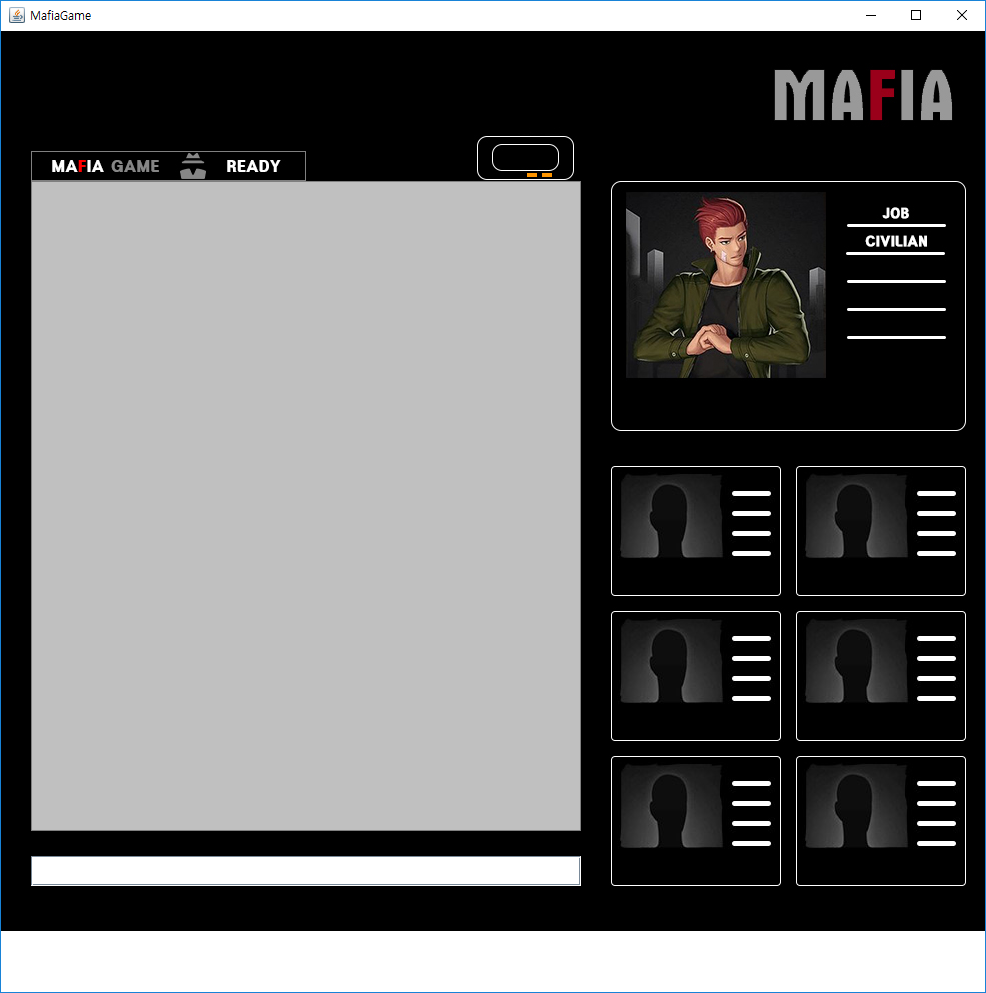
프로토타이핑

(회원가입 UI)

프로토타이핑

(로그인 UI)

(게임 방 UI)



Test Case



(수정 시 첨부파일 참고)

개발자테스트

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 마피아게임 Checklist | | | | | | | |
| No | 내용 | Yes | No | NP | 진행상황 | | 비고 |
| 설계 | 개발 | Yes = 적용됨  No = 해당 범위 외  NP = 추후 검토 |
| 1. 접속 정보처리 | | | | | | |
| 1 | 아이디의 제약조건 동작 | O |  |  | 완료 |  |
| 1 | 게임 설명 다이어로그의 확인 |  |  | O |  | 예정 |
| 2. 게임진행 | | | | | | |
| 1 | 밤 낮 쓰레드 구현 | O |  |  | 완료 |  |
| 1 | 투표 버튼에 따른 결과 반영 | O |  |  | 완료 |  |
| 1 | OUT 대상자 동작 정지 및 표기 변경 | O |  |  | 완료 |  |
| 1 | 게임 승리 or 패배 메시지 동작 | O |  |  | 완료 |  |
| 1 | 레디 버튼에 의한 게임 시작 | O |  |  | 완료 |  |

* 개발자 테스트 체크리스트 작성

통합테스트를 위한 테스트 시나리오 작성

* 테스트 시나리오 실무 작성

|  |  |  |  |
| --- | --- | --- | --- |
| 테스트 시나리오 ID | 테스트 시나리오 | 테스트 시나리오 상세설명 | 비고 |
| ITSC001 | 투표의 집계가 정확한가 | 투표 수에 따라 여러가지 경우의 수를 나누어 결과를 전송할 수 있도록 한다. |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 테스트케이스 실무 작성

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 테스트 시나리오  ID | 테스트  시나리오 | 테스트 케이스  ID | 테스트  케이스  설명 | 사전  조건 | 테스트  데이터 | 예상  결과 | 결과  확인  방법 | 실제  결과 | Pass  /  Fail |
| ITSC001 | 투표 결과 확인 | ITSC001  -01 | 투표 | 유저  접속 | 유저 및 직업 | 클릭 한 사용자에게 투표 | 화면 내 출력 | 이상 蕪 | P |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

* 테스트흐름도

(수정 시 첨부파일 참고)

* 통합테스트 문제해결 시나리오

|  |  |
| --- | --- |
| 평가 일자 : 2018. 06. 19 사번: 11111111 이름 : 이 도 협 | |
| 학습1) 개발자  테스트케이스  설계하기 | * 프로젝트의 특성을 반영한 테스트 케이스와 테스트 시나리오 작성 |
| 학습2) 개발자  통합테스트하기 | * 통합테스트 계획에 따른 테스트 수행 순서 설명 |
| 학습3) 개발자  결함조치 하기 | * 테스트 수행 결과에서 발견된 결함을 식별하고 조치 우선순위 결정 |
|  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 통합 테스트 결과 보고서 | | | | | | |
| 시스템/  서브시스템 | |  | 테스트  담당자 |  | 테스트  일자 |  |
| 테스트 케이스  ID | |  | 확인자 |  | 완료 일자 |  |
| 설명 | 테스트 케이스에 대한 간략한 설명 | | | | | |
| 테스트 시나리오 | | | | | | |
| 작업내용 | | | 작업구분/테스트데이터/관련 Use Case/테스트결과 및 오류내용 | | | |
| 작업 절차 #1.  테스트 시나리오를 위해 수행해야 할 작업내용 기술 | | | 작업 구분 | |  | |
| 테스트 데이터 | |  | |
| 관련 Use Case ID/명 | |  | |
| 테스트 결과 | | 오류 발생 시 오류 내용 | |
| 작업 절차 #2.  테스트 시나리오를 위해 수행해야 할 작업내용 기술 | | | 작업 구분 | |  | |
| 테스트 데이터 | |  | |
| 관련 Use Case ID/명 | |  | |
| 테스트 결과 | | 오류 발생시 오류 내용 | |
| 비고 | | | 테스트 세부 내용 기재(필요시) | | | |
| 최종 평가 | | | 합격 / 불합격 : | | | |
| 사유 : | | | |

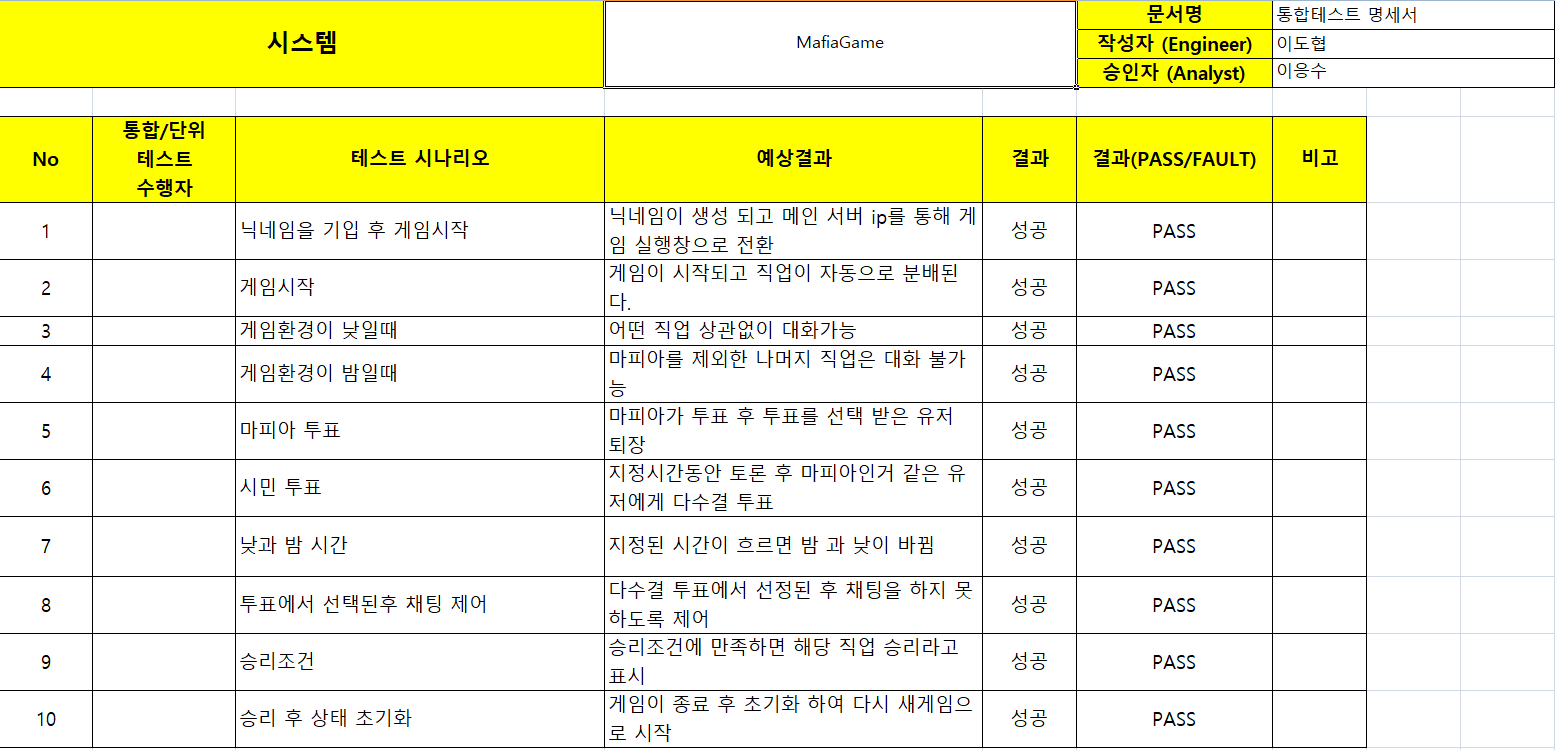
* 통합 테스트 결과 보고서
* 결함 관리 대장 작성

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 결함 관리 대장 | | | | | | | | | |
| 발견  일자 | 발견자 | 결함  구분 | 영향  (심각도) | 우선  순위 | 상세  내용 | 결합  유입  단계 | 결함  유형 | 결함  상태 | 결함  조치 |
| 결함  발견  일자 |  | 에러  결함 | A(Major)  B(Minor)  C(Trifle) | 즉시  주의  대기 |  | 결함  발견  단계 | 기획  설계  코딩  테스트 | 동작가능  동작불가 | 분석  수정  완료 |
| 2018.  08.  29 | 이도협 | 에러 | B | 즉시 | OUT 이미지 경로 에러 | 발견 | 코딩 | 동작가능 | 수정 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

* 형상관리 산출물 작성

|  |  |  |  |
| --- | --- | --- | --- |
| **형상관리산출서** | | | |
| 프로젝트명 | 네트워크를 활용한 마피아 게임 | | |
| 산출요청자 | 이도협 | 요청일자 | 2018. 6. 19 |
| 항목 ID | PR\_10(프로토콜관리계획)\_01 | 항목명 | 프로토콜관리계획서 |
| 처리담당자 |  | 완료예정일자 |  |
| 승인자 | 이 도 협(인) | 처리결과확인 | 이 응 수(인) |
| 요청내용 | 1. 코드의 일관성 부여   \*기존의 프로토콜의 복잡성에 의한 관리의 어려움에 대응하기 위하여 프로토콜에 일관성부여.   1. 코드 식별의 용이성 부여   \* 기존에 산재되어있던 코드를 swich~case문을 통하여 모아준다. | | |
| 변경사유 | 많은 양의 프로토콜관리의 어려움 보완 | | |
| 처리결과 | 변경요청사항의 내용이 이상 없이 보완됨. | | |

통합테스트 명세서



정보시스템이행

모니터링 관련 문서…

제품 소프트 웨어 패키징

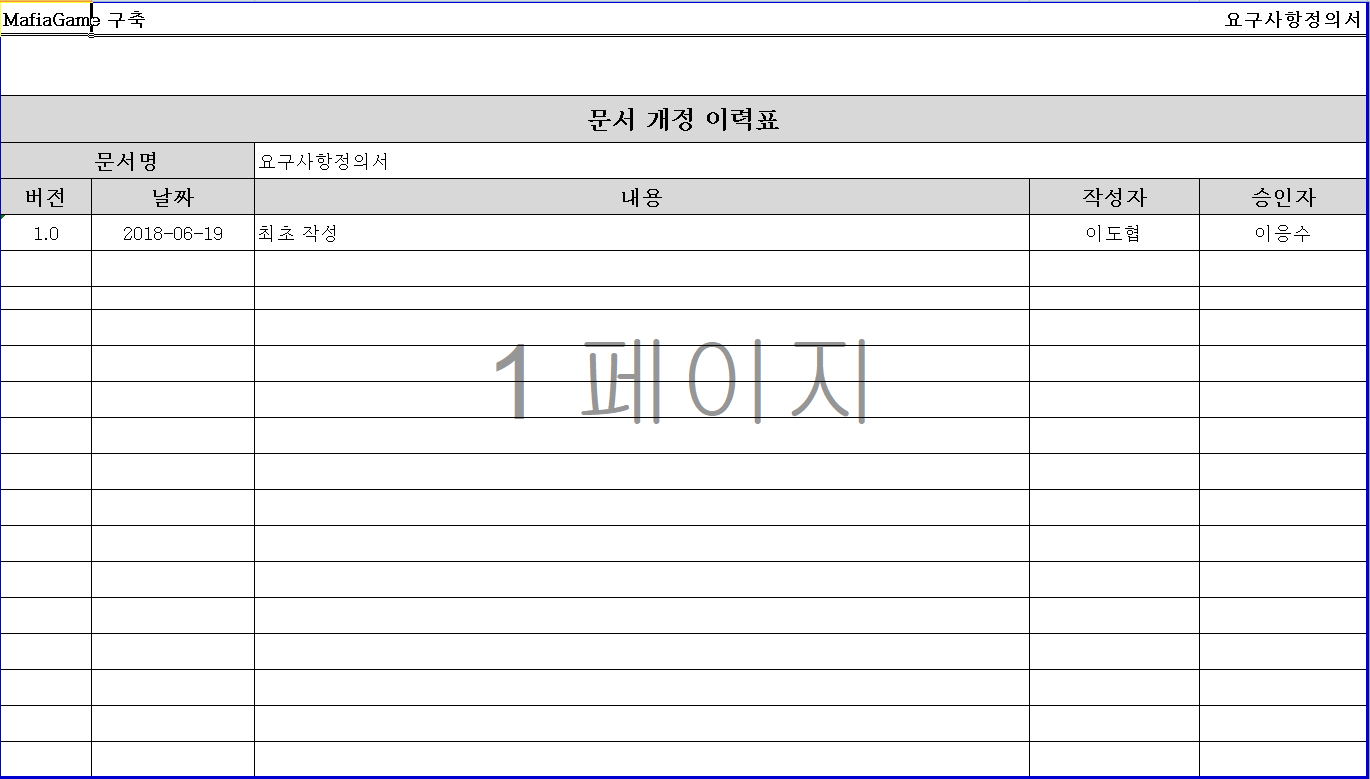
1. 제품 소프트웨어 패키징하기

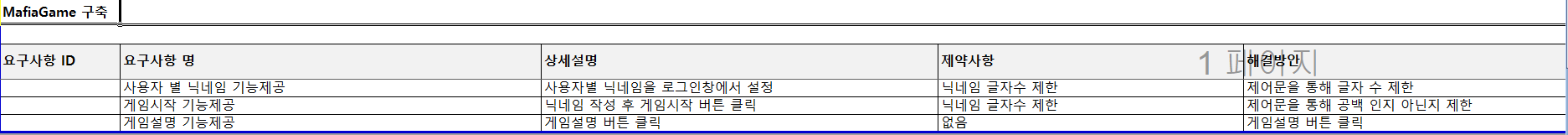
- 사용자 중심의 제품 소프트웨어 패키징

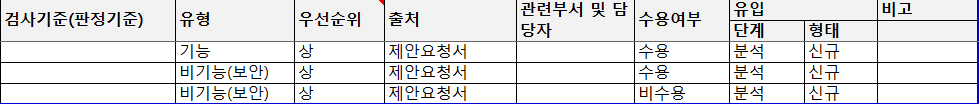
|  |  |
| --- | --- |
| **신규 개발 소스의 목적 및 기능** | |
| 1. 기능 수행을 위한 입출력 데이터 : 소스 기능을 통해 처리되는 int, char형 데이터이다.  2. 전체적인 기능 정의 및 데이터 흐름 : 소스 내부 메인 함수의 기능 식별, 관련 데이터  의 흐름 및 출력 절차를 확인한다.  3. Function 단위 및 Output에 대한 상세 정의 : 메인 함수 이외의 호출 함수를 정의하고 이에 대한 출력값을 식별한다. | |
| **모듈 단위 분류 및 모듈화** | |
| 1. 수행을 위한 기능 단위 및 서비스 분류 : 모듈로 분류할 수 있는 기능 및 서비스 단위  로 분류한다.  2. 패키징 개념에서의 기능의 공유와 재활용 분류 : 여러 번 호출되어 지는 공유, 재활용  함수를 분류한다.  3. 모듈 간 결합도와 응집도 (Loose Coupling & Strong Cohesion) 식별 : 모듈화를 위해 결  합도와 응집도를 식별해내고 모듈화 수행을 준비한다. | |
| **단위별 빌드 진행** | |
| 1. 신규 개발 소스 및 컴파일 결과물 준비 : 빌드 진행을 위한 신규 소스 및 결과물을 준  비한다.  2. 정상 기능 단위 및 서비스 분류 : 정상으로 빌드가 되는 기능 및 서비스를 사전에 분류한다.  3. 빌드 도구 확인 및 정상 수행 : 빌드 도구의 사전 선택 확인 및 빌드 도구를 통한 빌드를 수행한다.  4. 컴파일 이외의 도구의 다양한 기능 확인 : 컴파일 이외의 에디터, CI 도구 등 관련 도구들에 대한 추가 학습을 진행한다. | |
| **사용자의 요구 사항 및 사용 환경** | |
| **사용 환경**  **사용 환경** | EMB00000b844683 |
| **사용자 테스트**  **결과** | 2. 모듈 단위의 여러 가지 기능별 사용자 환경 테스트 : 수행한 모듈화의 결과에 따른 여러 가지 사용자 테스트를 수행한다. |
| **사용자 건의 사항** | |
| 채팅 간 시스템 메시지의 식별이 명확하지 않음 | |
| **변경 및 개선 사항** | |
| 1. 변경점  - 더 이상 UI 상 생존자가 사망자로 사망자가 생존자로 표시되지 않음  - 타이머 UI 추가로 사용자의 편의성 향상 | |

요구사항정의서

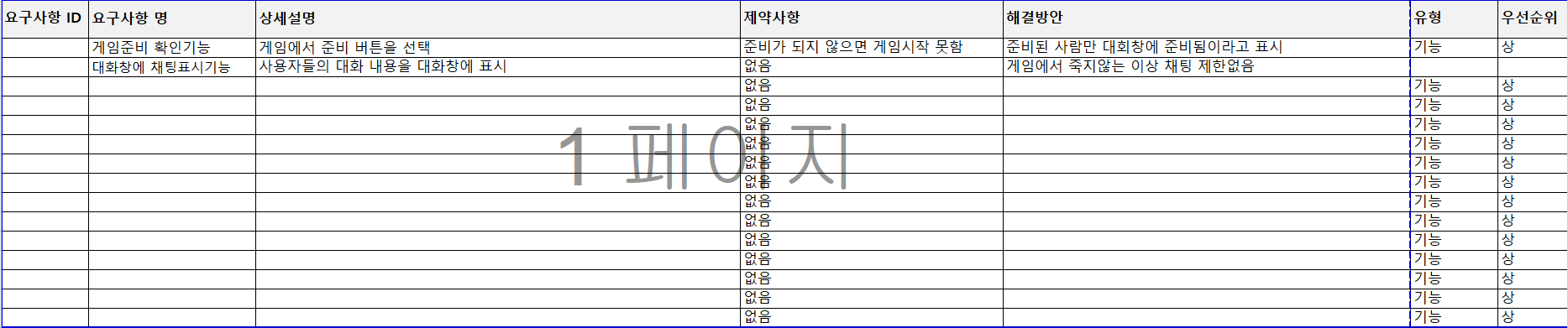
1.개정이력



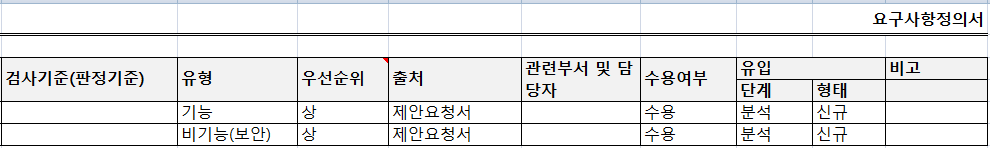
2.공통



3.사용자



4.관리자



- 버전을 고려한 제품 릴리즈 노트 작성하기

|  |  |
| --- | --- |
| **릴리즈 노트** | |
| **Header** | 문서 이름 : rnote\_001  제품 이름 : 마피아 게임  버전 번호 : 1.01a  릴리즈 날짜 : 2018.06.15.  노트 버전 : 1.00 |
| **개요** | 마피아 게임 중 발생한 오류에 대한 수정 |
| **목적** | 1. 기존의 버전에서 발생한 오류에 대한 수정  2. 타이머 UI 기능 추가 |
| **이슈 요약** | 게임 중 자신의 생존 상태 이미지가 반대로 출력 |
| **재현 항목** | 1. 서버 동작  2. 클라이언트 실행 후 참가자 6명 접속 확인  3. 게임 시작 후 본인 상태 UI 이미지 확인  4. 게임 진행 및 OUT 참가자 UI 이미지 확인 |
| **수정/개선** | 1. 기존의 버전에서 발생한 오류에 대한 수정  EMB00000b844684[p\_1 게임 실행 중 화면]  [p\_1]의 우측 상단 화면이 생존으로 표시되어야 함에도 DEAD로 출력  EMB00000b844685  [p\_2 오류 코드 화면]  [p\_2]의 코드 중 setIcon();내의 변수명이 바뀌어 입력되어 있음을 확인 후 수정  2. 타이머 UI 기능 추가  EMB00000b844686[p\_3. 타이머 UI]  [p\_3]와 같이 타이머 부분을 추가한 UI 제공으로 보다 직관적으로 남은 시간을 알 수 있음  추가 된 소스는 다음 [p\_4]와 같음  EMB00000b844687[p\_4 타이머 UI 추가] |
| **사용자 영향도** | 오류 수정으로 자신의 상태를 정확하게 파악 가능  채팅창 내에서만 확인 가능하던 타이머를 UI에서 보다 직관적으로 확인 가능하여 게임에 대한 몰입도 상승 |
| **SW 지원 영향도** | - |
| **노트** | 실행 환경 :  EMB00000b844683실행 툴 : Eclipse Java EE IDE Web Developers. Oxygen.3a Realease (4.7.3a) |
| **면책 조항** | - |
| **연락 정보** | 협조 02-0000-0000 |

- 패키징 도구를 활용한 설치, 배포 수행하기

|  |  |
| --- | --- |
| **패키징 수행 결과 보고서** | |
| 릴리즈 노트 | rnote\_001 |
| 패키징 도구 선택 | - |
| 암호화 보안 기능 | 암호화(Encryption)  -콘텐츠 및 라이선스를 암호화하고, 전자 서명을 할 수 있는 기술 -PKI, Symmetric/Asymmetric Encryption, DiGital Sinature  키 관리 (Key Manangement)  -콘텐츠를 암호화한 키에 대한 저장 및 배포 기술 (Centralized, Enveloping)  암호화 파일 생성 (Packager)  -콘텐츠를 암호화된 콘텐츠로 생성하기 위한 기술 -Pre-packaging, On-the-fly Packaging  식별 기술 (Identification)  -콘텐츠에 대한 식별 체계 표현 기술 -DOI, URI  저작권 표현 (Right Expression)  -라이선스의 내용 표현 기술 -XrML/MPGE-21 REL, ODRL  정책 관리 (Policy management)  -라이선스 발급 및 사용에 대한 정책표현 및 관리기술 -XML, Contents Management System  크랙 방지 (Tamper Resistance)  -크랙에 의한 콘텐츠 사용 방지 기술 -Code Obfuscation, Kernel Debugger Detection, Module Certification -Secure DB, Secure Time Management, Encryption  인증(Authentication)  -라이선스 발급 및 사용의 기준이 되는 사용자 인증 기술 -User/Device Authentication, SSO, DiGital Certificate |
| DRM 흐름 | EMB00000b844688 |
| 암호화/보안 기능 적용 확인 |  |
| 암호화/ 보안 기능 체크리스트 |  |

제품 소프트웨어 패키징

1. 제품 소프트웨어 설치 매뉴얼
2. 개요

|  |  |  |
| --- | --- | --- |
| 제품명 | 네트워크를 이용한 마피아 게임 | |
| 목적 | * 네트워크 게임 플레이 환경 제공 * 채팅 기능 제공 | |
| 기능 | 접속 | 이름과 IP 주소를 통해 서버에 접속 |
| 게임방법 | 다이어로그를 통해 게임 도움말 출력 |
| 채팅 | 동시 접속자와 실시간 채팅 가능 |
| 게임 플레이 | 6인용 마피아게임 |
| 낮과 밤 | 쓰레드를 통한 낮과 밤 구현 밤에는 마피아만 채팅 가능 |
| 역할 | 마피아와 시민 존재 |
| 게임 반복 | 게임 종료 후 레디버튼으로 게임 반복 가능 |
| 시스템메세지 | 사회자 역할을 시스템이 수행 |
| 투표시스템 | 범인을 찾기 위한 투표 가능 |

1. 문서 정보

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 버전 | 작성자 | 작성일 | 검토자 | 일시 | 검수인 |
| v1.01 | 이도협 | 2018-06-19 | 이도협 | 2018-06-19 | 이도협 |
| 변경 내용 | 최초 작성 | | | | |

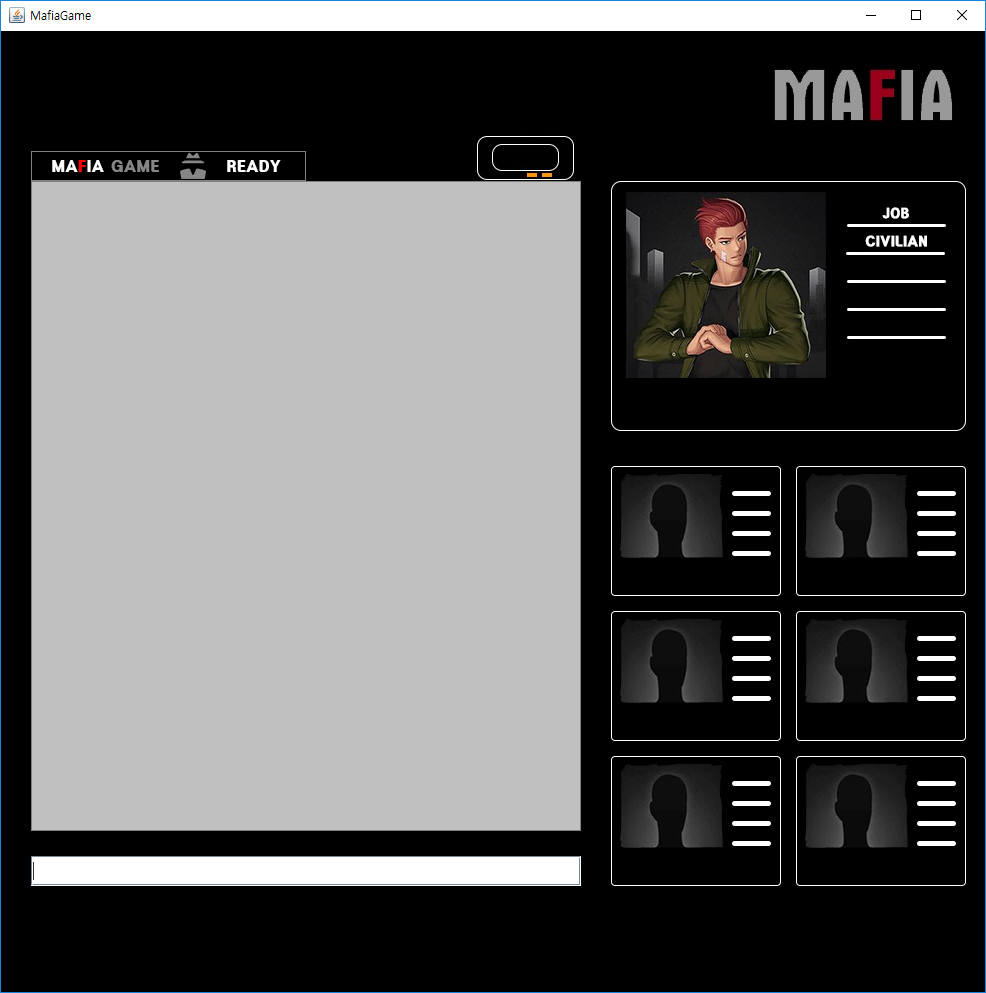
1. 설치 방법
2. 설치 : 배포된 JAR 파일을 실행한다.

Cf) 정상 설치 시 실행 화면

* 로그인 화면



* 메인 화면



1. 삭제 : JAR 파일을 제거한다.
2. 제품 소프트웨어 사용자 매뉴얼
3. 문서 개요

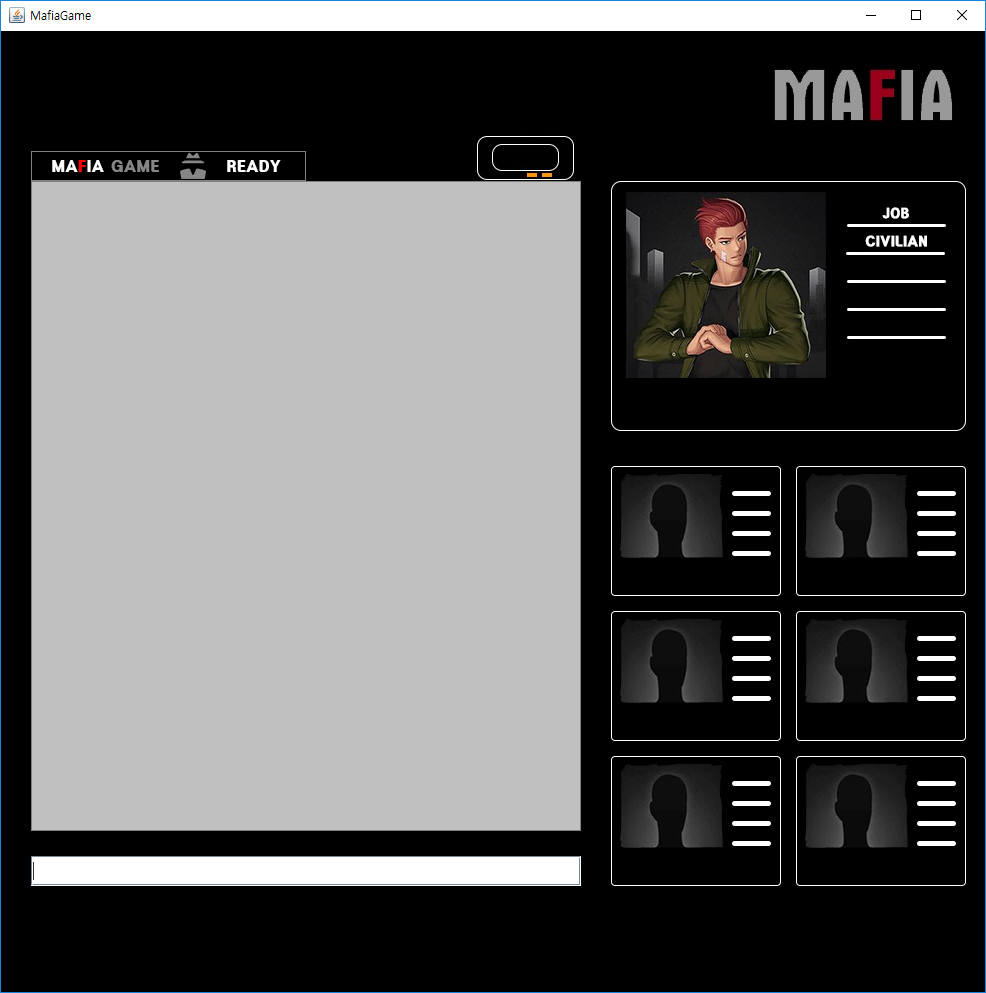
|  |  |
| --- | --- |
| 제품 명칭 | 네트워크를 활용한 마피아 게임 |
| 제품 버전 | v1.01 |
| 문서 번호 | BDOC1001 |
| 제품 시리얼 번호 | KRB0010928 |
| 구입 날짜 | (공란) |

1. 기능별 제품 이용 가이드
2. 로그인

* 제품 실행 후 이름을 입력하면 서버로 접속 할 수 있습니다.
* 접속 화면

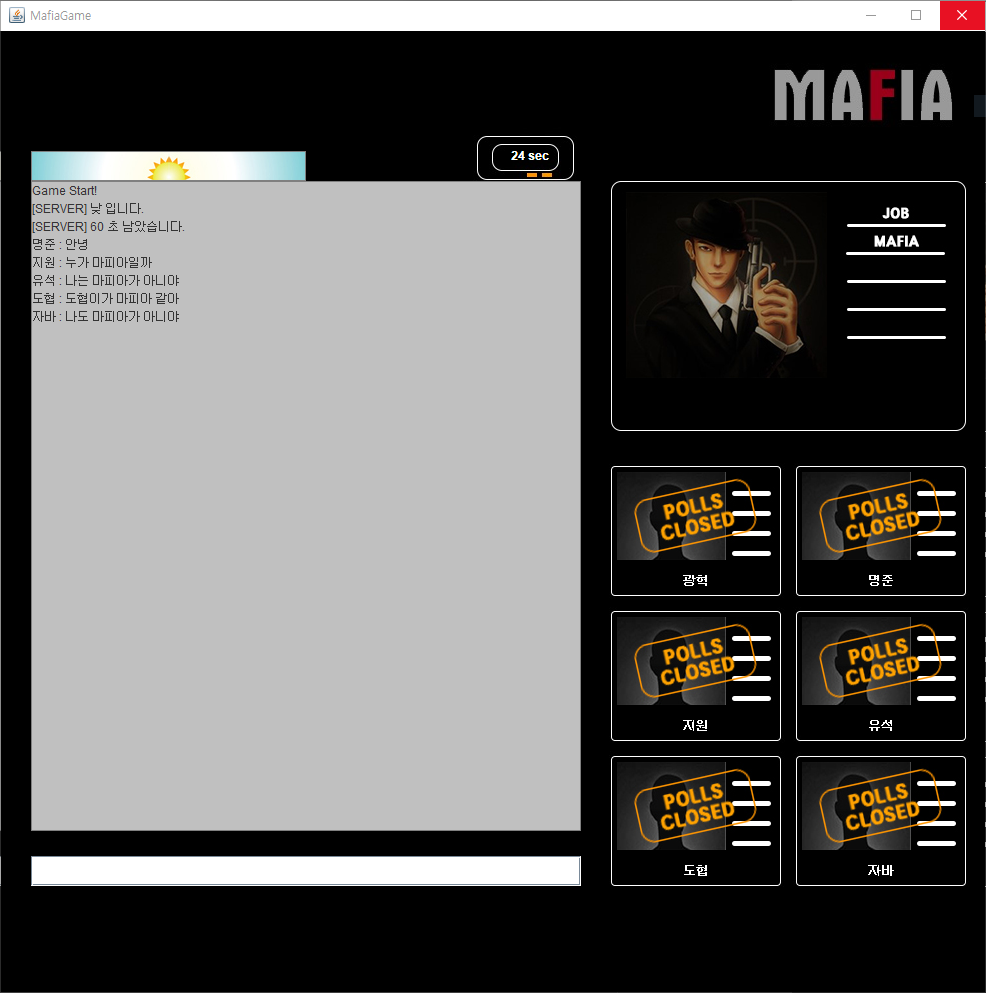


* 로그인 성공 시 메인 화면



1. 게임 플레이

* 모든 플레이어가 READY 버튼을 누르면 자동으로 게임을 시작할 수 있습니다. 게임 시작 후 사용자 별로 시민과 마피아가 정해지며 낮부터 시작하여 채팅이 활성화 됩니다.
* 낮과 밤이 일정 시간을 가지고 타이머가 동작하며 낮이 끝나면 전체 인원이 투표를 하여 마피아를 선정합니다.밤의 경우 마피아만이 채팅이 가능하며 마찬가지로 일정 시간이 지나고 마피아만이 살해할 시민을 선정합니다.
* 게임 화면

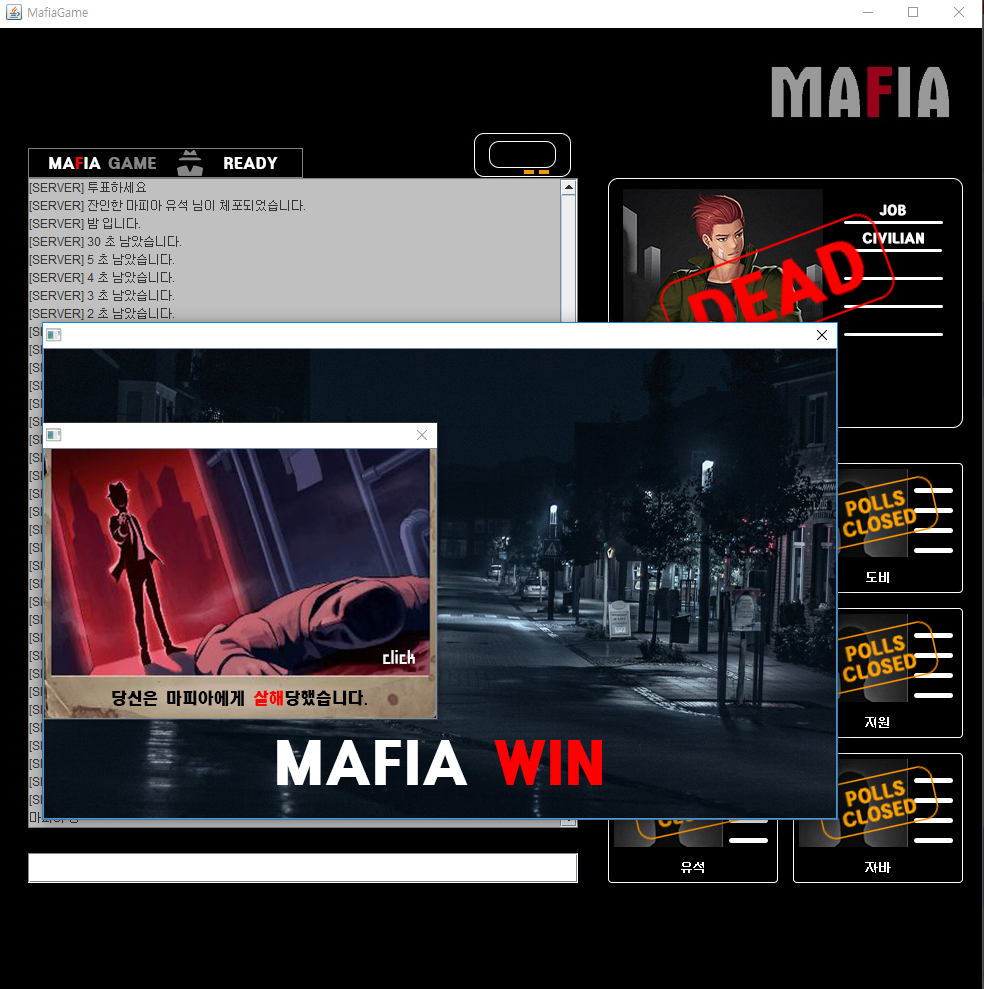


* 마피아가 시민의 숫자와 같거나 마피아가 모두 검거되면, 게임이 종료됩니다.
* 승리 화면



1. 투표 기능

투표 결과에 따라 사용자는 체포 될 수 도 사망할 수 도 있습니다.



1. 주의 사항
2. 접속 전 서버의 IP주소를 미리 확인해 클라이언트에 입력해 주십시오.
3. 접속 에러가 발생한 경우 클라이언트 프로그램을 종료 후 다시 실행하십시오.
4. 접속 에러가 지속될 경우 서버의 정상 작동 여부를 확인해 주십시오.
5. 게임 플레이 중간에 접속을 종료할 경우 게임 플레이에 오류가 발생할 수 있습니다.