



2023 분산 프로그래밍

보험사 소프트웨어 개발 프로젝트

기획서

Team iGo

- 김건우
- 김나연
- 김민수
- 유병규
- Utku Gulbardak

목차

소개.....	3
1. 설계서 개요.....	3
2. 프로젝트 개요.....	3
3. 서약서.....	4
요구사항 수집.....	5
1. 기능 요구사항.....	5
2. 비기능 요구사항.....	7
도메인 분담 및 설계 & 개발 관련 규정 수립.....	8
1. 세부 도메인 분담.....	8
2. 사전 규정.....	9
1. 사용할 기술, 커뮤니케이션 툴.....	9
분석 : Usecase Diagram.....	10
1. Overall Usecase Diagram.....	10
Scenarios.....	11
설계 : Class Diagram.....	67
1. Overall Class Diagram.....	67
2. Class Description.....	68
맺으며.....	103
팀원 별 소감.....	103
앞으로의 방향성.....	104

소개

1. 설계서 개요

본 설계서는 소프트웨어 설계원칙을 준수하고, 설계된 구조에 대한 설명과 앞으로의 개발 방향성을 도출하여 우리 팀에게 보험사 소프트웨어 개발을 의뢰한 고객사에게 시각적, 기능적으로 모호함이 없는 제품의 설계 및 초안을 제시하는 것을 목적으로 한다. 즉, 본 기획서는 제품에 대한 설명을 통해 고객의 요구사항을 명확히 하고, 기능적 및 객체 지향적으로 설계하여 해당 설계를 고객에게 납득시키며, 후에 특정 부분의 수정 등을 요청받을 시 고객의 요구에 발 빠르고 유연하게 대처할 수 있도록 하기 위함이다.

2. 프로젝트 개요

2023년 분산 프로그래밍 강의에서는 2007년에 역사속으로 사라진 '신동아화재'사가 2005 년도에 보험사 소프트웨어 개발을 요청했던 RFP를 참고자료로서 제공한다.

신동아화재 NSP IT 시스템구축 제안 요청서(RFP)

신동아화재
NSP TFT

위는 그 RFP의 겉표지이며 해당 문서의 총 분량은 49장에 달한다. 이 문서를 분석하고, 보험사의 비즈니스 로직과 비즈니스 용어 등 관련 도메인 지식을 학습하여 보험사 시스템을 설계하고 개발한다. 즉, 우리는 가상의 기업(고객)이 '아이고(iGO)' 팀에게 RFP를 건네주며 시스템 구축을 의뢰한 상황을 가정하고 작업에 착수한다. 우리는 해당 프로젝트를 통해 소프트웨어 아키텍처 설계 능력과, 개발 능력뿐 아닌 도메인

지식의 중요성을 몸으로 체득하고, 이 경험을 토대로 후에 실제 비즈니스 환경에서 보다 능숙하게 문제를 해결해 나갈 수 있을 것으로 기대하며 프로젝트에 임한다.

3. 서약서

Pledge

1. 본인은 보고서 및 논문의 내용을 표절하지 않겠습니다.
2. 본인은 타 보고서 및 논문의 내용을 인용할 시 인용 형식을 준수하고 출처를 표기하겠습니다.
3. 본인은 보고서 및 논문 대필을 청탁하거나 청탁 받지 않겠습니다.
4. 본인은 기타 보고서 및 논문 작성과 관련된 위법 행위를 포함한 어떠한 부정한 행위를 하지 않겠습니다.

본인은 위의 네 가지 조항을 준수할 것이며, 보고서 및 논문에 어떠한 거짓이나 허위사실도 기입하지 않고, 성실히 작성할 것을 맹세합니다.

2023년 04월 25일

김건우



김나연



김민수



유병규



Utku Gulbardak



요구사항 수집

1. 기능 요구사항

아래는 시스템에서 구현되어야할 기능 요구사항이다.

1. 고객 정보 관리 기능

- 시스템은 고객의 이름, 성별, 나이, 연락처 등 기본적인 신상정보를 저장하고, 보험사의 직원들은 이를 조회할 수 있어야 한다.
- 만일 보험에 가입한 고객이나 가입을 희망하는 고객의 경우 직원은 시스템을 통해 고객들에게 건강검진 정보, 직업, 가족력 등 보험료나 보험 가입 여부에 영향을 미치는 추가적인 정보를 요청 및 조회할 수 있어야 하며, 시스템은 해당 정보들을 저장할 수 있어야 한다.

2. 직원 정보 관리 기능

- 시스템은 직원의 이름, 성별, 나이, 연락처, 직급 등 신상정보를 저장할 수 있어야 한다.

3. 상품 개발 & 출시 기능

- 상품 개발팀 직원은 시스템을 통해 개발하려는 상품의 현재 진행 단계를 저장할 수 있어야 한다.
- 상품 개발팀 직원은 시스템을 통해 해당 상품을 최종적으로 출시하기 전에 컴플라이언스 팀에게 해당 상품의 세부 정보를 전송할 수 있어야 하고, 컴플라이언스 팀은 시스템을 통해 해당 정보를 받아 이를 검토하여 상품 개발팀 직원에게 해당 상품 출시를 허가 혹은 거절할 수 있어야 한다.

4. 광고 등록 기능

- 마케팅팀 직원은 효율적인 광고 제작을 위해 시스템을 통해 고객들을 대상으로 한 설문조사를 할 수 있어야 한다.
- 마케팅팀 직원은 시스템을 통해 보험 상품에 대한 광고의 정보를 기재하고, 이를 등록할 수 있어야 한다.

5. 상품 영업 & 판매 기능

- 영업팀 직원은 고객에게 적합한 상품을 영업(판매, 추천)하기 위해 시스템을 통해 고객의 정보를 조회할 수 있어야 한다.
- 영업팀 직원은 시스템을 통해 영업한 보험을 계약하고 싶어하는 고객에게 상품을 판매할 수 있어야 한다.

6. 계약 관리 기능

- 계약관리팀 직원은 시스템을 통해 고객이 가입한 보험 계약을 관리할 수 있어야 한다. 구체적으로, 보험료 미납 고객 조회 및 안내, 계약 만기 임박 고객 조회 및 안내를 할 수 있어야 한다.
- 고객이 시스템을 통해 실효 계약에 대해 보험 부활을 요청할 수 있어야 하며, 계약관리팀 직원은 시스템을 통해 고객에게 부활 심사 현황과 결과를 안내할 수 있어야 한다.

7. 보험료 납입 기능

- 고객은 시스템을 통해 본인이 가입한 보험 내역과 보험료 납입 내역을 확인할 수 있어야 한다.
- 고객은 시스템을 통해 보험료를 납입(결제)할 수 있어야 한다.

8. 재보험 기능

- U/W팀 직원은 시스템을 통해 재보험 계약시 발생할 손해율을 계산할 수 있어야 한다.
- U/W팀 직원은 시스템을 통해 재보험 회사와 재보험 계약을 체결하고 계약 내용을 저장할 수 있어야 한다.

9. U/W(인수심사) 기능

- U/W팀 직원은 시스템을 통해 보험 인수 시 발생할 손해율을 계산할 수 있어야 한다.
- U/W팀 직원은 시스템을 통해 인수심사 대상 리스트를 확인하여 해당 인수심사에 필요한 고객정보, 보험정보를 확인할 수 있어야 하며 은행담당원을 통해 고객 신용 정보를 받아올 수 있어야 한다. 또한, 금융감독원의 승인 및 보험 (재)인수심사를 진행할 수 있고 (재)인수심사 결과를 저장할 수 있어야한다.

10. 사고 접수 기능

- 고객은 사고 발생시 시스템을 통해 사고 접수 기능을 이용할 수 있어야 하며, 시스템은 사고 접수팀에게 사고가 접수되었음을 알리고 접수한 고객의 정보를 전달해야 한다.
- 사고 접수팀 직원은 시스템을 통해 사고를 접수받으면 해당 접수에 대한 수락/거절을 선택할 수 있어야 하며, 긴급한 상황일 경우 현장 조사팀 직원을 출동시킬수 있어야 한다.
- 사고(사건) 접수가 수락되었다면 시스템은 이를 보상 처리 팀에게 전달하여 해당 사건에 대한 보상 업무가 이루어질 수 있도록 해야한다.

11. 보상 처리 기능

- 보상 처리팀 직원은 시스템을 통해 보상이 필요한 고객과 사건 정보를 검토하고, 보상 여부를 판별하거나 보상금 책정 등 보상 처리와 관련된 일련의 업무를 수행할 수 있어야 한다.
- 보상 처리팀 직원은 시스템을 통해 해당 사건의 보상 처리 과정이 어디까지 진행되었는지 확인할

수 있어야 한다.

12. 구상 기능

- 보상 처리팀 직원은 시스템을 통해 보상금을 지급하고, 해당 사건의 손괴자 여부를 파악하여 손괴자가 존재하는 사건에 대해 손괴자의 정보를 조회할 수 있어야 한다.
- 보상 처리팀 직원은 손괴자가 합당한 구상 청구를 거절하는 경우 시스템을 통해 법무팀에게 소송을 요청할 수 있어야 하며, 법무팀은 이러한 요청의 발생을 시스템을 통해 확인할 수 있어야 한다.

2. 비기능 요구사항

아래는 시스템이 갖추어야 할 비기능 요구사항이다.

성능(Performance) :

- 서비스 응답 시간은 평균적으로 2초, 최대 7초 이내로 이루어져야 한다.

보안(Security) :

- 사용자의 정보와 관련된 데이터는 누구에게나 공개되어서는 안된다.
- 액세스 권한이 필요한 서비스는 해당 권한이 없는 사용자가 접근할 수 없어야 한다.

가용성(Availability) :

- 서비스는 항상 운영되어야 한다.

신뢰성(Reliability) :

- 서비스는 사용자에게 정확한 정보를 제공해야 하며, 만일 잘못된 정보가 제공되거나, 해당하지 않는 고객에 대해 요금이 부과되거나 결제가 이루어지는 등의 경우가 없어야 한다.
- 만일 위에서 언급한 문제들이 발생하면 최대 1시간 이내로 해결되어야 한다.

사용성(Usability) :

- 본 소프트웨어를 사용하는 사용자는 보험사의 고객들과 보험사 직원들이기에, 해당 소프트웨어는 이 양측의 사용자 모두에게 직관적이고, 사용하기 쉬운 형태여야 한다.

확장성(Scalability) :

- 서비스는 사용자의 수가 평균 사용자 수에서 크게 증가해도 오류나 서비스 마비 없이 동일한 성능으로 동작해야 한다.

이식성(Portability) :

- 서비스는 사용자의 기기의 종류에 상관없이 동일한 기능을 제공해야 한다.

호환성(Compatibility) :

- 서비스는 사용자의 운영체제에 상관없이 동일한 기능을 제공해야 한다.

도메인 분담 및 설계 & 개발 관련 규정 수립

1. 세부 도메인 분담

보험사의 비즈니스 영역은 광범위하고 복잡하기에, 아래와 같이 각각의 팀원들은 세부 도메인을 맡아 해당 파트의 지식과 용어를 깊게 학습하여 본인의 도메인 영역의 비즈니스 로직을 숙지한다. 맡은 도메인 영역을 기반으로 유스케이스 다이어그램과 클래스 다이어그램을 설계하고, 후에 이 설계를 바탕으로 맡은 도메인의 코드 개발 또한 수행하게 된다.

이름	담당 도메인
김건우	사고 접수, 보상 처리 관련
김나연	계약 관리, 보험료 납부 관련
김민수	상품 판매, 영업 및 마케팅 관련
유병규	인수심사(U/W), 재보험 관련
Utku Gulbardak	상품 개발, 컴플라이언스 관련

2. 사전 규정

1. 사용할 기술, 커뮤니케이션 툴

요구사항 명세화 및 설계



Enterprise Architect

기획서 등 문서 작성



Google Docs

형상 관리



회의 등 커뮤니케이션



slack

소프트웨어 개발



eclipse

데이터베이스

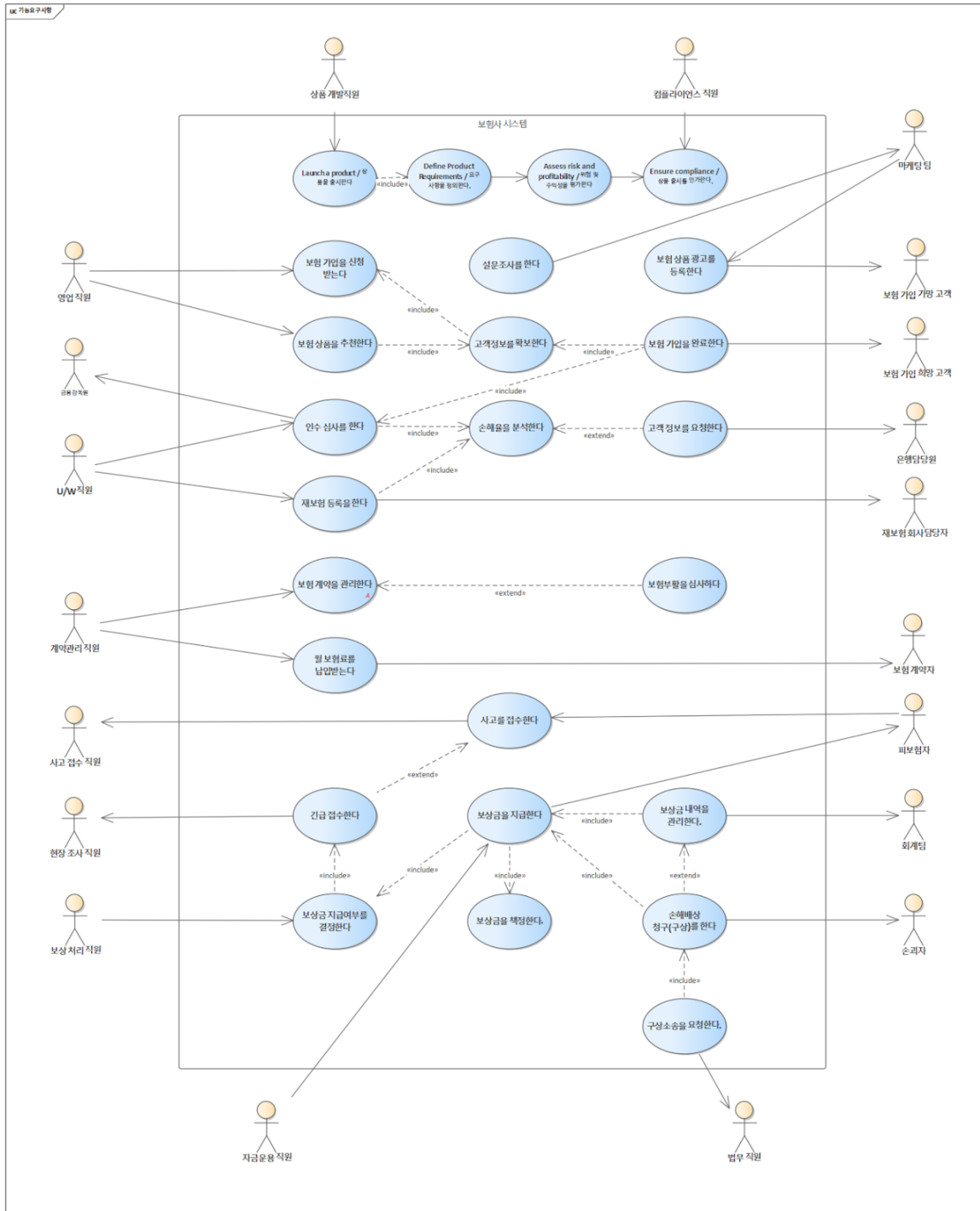


* 이번 프로젝트에서는 설계 단계에선 Sparx사의 Enterprise Architect를 사용하고, 구현단계에서는 Spring Framework, JPA를 사용하지 않고 순수 JAVA와 JDBC를 통해 DB를 연동하여 본질적이고 원론적인 부분을 학습하는것이 제약조건이기에 이를 준수한다.

분석 : Usecase Diagram

1. Overall Usecase Diagram

아래는 시스템 구축을 요청한 고객의 기능 요구사항을 도식화한 유스케이스 다이어그램이다.



Scenarios

시나리오는 각각의 유스케이스 내에서 시스템이 어떻게 기능하는지에 대해 직관적으로 그 흐름을 기술한 것을 말한다. 유스케이스 내에서 일어나는 정상적이고, 기본적인 플로우를 **BasicPath**, Basic Path에 비해 적은 경우의 수로 발생하는 플로우는 **Alternate(A)**로, 해당 유스케이스에서 발생할 수 있는 예외(오류) 플로우는 **Exception(E)** 으로 정의한다. E1-E10이나 A1-A1 등의 표기는 해당 E1 플로우 중에 발생할 수 있는 Exception 속의 Exception 이나 A1플로우 중에 발생할 수 있는 Alternate 속 Alternate를 나타낸다.

테스트를 진행하기 앞서 : JAVA 프로젝트 파일 내에 README.md 파일을 읽고 ConnectorJ 연동, DB생성, 테이블 생성, 더미데이터 추가하고 아래 유스케이스들을 테스트 해야한다.

-상품 개발, 컴플라이언스 파트 (Utku Gulbardak)

Use Case Name : Define Product Requirements (요구 사항을 정의한다.)

Description : *This Use Case is where the first steps of the product are defined and start of creating a prototype process. (이 사용 사례는 제품의 첫 번째 단계를 정의하고 이해 관계자와 논의하고 프로토타입 프로세스를 만들기 시작하는 곳입니다.)*

- **Basic Path**

1. The product development team employee login with entering ID into a text box. (E1)

Test : Choose 2 in the login page (employee login) and enter "pd2025" as ID number for Product Development Team. Then select 1 (Define new prototype) and fulfill the necessary parts and enter "yes" to finish creating a new prototype. Prototype status is "pending", it means that the compliance team should check it. Log out and log in as a compliance team employee.

테스트 : 로그인 페이지(사원 로그인)에서 2를 선택하고 상품개발팀 아이디로 "pd2025"를 입력합니다. 그런 다음 1(새 프로토타입 정의)을 선택하고 필요한 부분을 채우고 "예"를 입력하면 새 프로토타입 생성이 완료됩니다. 프로토타입 상태는 "보류 중"이며 규정 준수 팀에서 확인해야 함을 의미합니다. 로그아웃하고 준법감시팀 직원으로 로그인합니다.

1. 상품개발팀 직원이 입력란에 아이디와 비밀번호를 입력하여 로그인한다. (E1)

2. System shows"Welcome + UserName).

2. 입력이 잘못된 경우 시스템에 "Welcome + UserName" 또는 "잘못된 ID/비밀번호"가 표시됩니다.

3. The product development team employee creates a new prototype by pressing the "define new prototype" button.

3. 제품개발팀 직원이 "신규 시제품" 버튼을 눌러 새로운 시제품을 제작합니다.
4. System opens a new page and asks to fill in the necessary information. Such as "Product name" , "product ID" , "Description", "Requirements" as text boxes and "category" as a check box. Check box has options as "Life Insurance", "Damage Insurance" "Saving Insurance".
4. 시스템이 새 페이지를 열고 필요한 정보를 입력하도록 요청합니다. "제품 이름", "제품 ID", "설명", 텍스트 상자의 "요구 사항" 및 확인란의 "제품 카테고리"와 같은 것입니다. 생명보험,손해보험,저축보험 중에 카테고리를 선택합니다.
- 5.After checking the category, "detailed category" is displayed as a check box. If you checked "Damage Insurance" as a category, the Check box has detailed category options as "car insurance", "fire insurance", "marine insurance", "travel insurance", "fire insurance" and "other". If "other" is selected, It opens a text box to fulfill. If you check "Life Insurance", the detailed category shows "cancer insurance", "death insurance" and "other". If you check "Saving Insurance", the detailed category shows "pension insurance", "saving insurance" and "other".
5. . 카테고리를 선택했다면, 이후에 디테일한 카테고리를 선택하는 확인란이 표시됩니다. 만일 "손해 보험" 카테고리를 선택했다면 디테일 카테고리에는 "자동차", "사고", "여행", "해상", "화재" 및 "기타" 옵션이 있습니다. "기타"를 선택하면 채워야 할 텍스트상자가 나타납니다. 만일 "생명보험" 카테고리를 선택했다면 디테일 카테고리에는 "암보험", "사망보험" 및 "기타" 옵션이, "저축보험"을 선택했다면 "연금보험", "저축보험" 및 "기타" 옵션이 표시됩니다.
6. After definition is complete, the system automatically ask to The product development team if they confirm the prototype.
 - If "yes" pressed, "New prototype created" message shows up and the Compliance Development Team gets a notification about it and they have been asked to review and confirm the prototype.
 - If the answer is "no", the system goes back to the main product development team menu.
7. If there are any update suggestions from the compliance team, prototype status changes to "need update" so The product development team updates the task as it is recommended and presses the "update prototype status" button. and prototype status changes back to "pending" again. This loop keep happening until Compliance Team confirms the prototype.(A2,A3)

8. Once the compliance team changes the prototype status to “confirmed”, then a Risk Report has to be performed.

- A1, If compliance team requires more information (이해 관계자가 더 많은 정보를 요구하는 경우)

1. A compliance team member responds to the task with a suggestion that requires further clarification.

Test : Choose “ provide Feedback “ option from the Compliance Team menu and enter a feedback. (prototype status changes to “need update”) Then go log out and log in again as product Development Team employee and choose option 0. Show prototypes and see the feedback. Then go back and choose option 3, Update prototype and choose what needs to be updated and enter new information. prototype status again turns to “pending”. Then go back to the Compliance Team and choose “review Prototype” and enter “confirm” so the update is finished.

테스트 : 컴플라이언스팀 메뉴에서 "피드백 제공" 옵션을 선택하고 피드백을 입력합니다. (프로토타입 상태가 "업데이트 필요"로 변경됨) 그런 다음 로그아웃하고 제품 개발 팀 직원으로 다시 로그인하여 옵션 0을 선택합니다. 프로토타입을 표시하고 피드백을 확인합니다. 그런 다음 돌아가서 옵션 3, 프로토타입 업데이트를 선택하고 업데이트해야 할 항목을 선택하고 새 정보를 입력합니다. 프로토타입 상태가 다시 "대기 중"으로 바뀝니다. 그런 다음 컴플라이언스 팀으로 돌아가서 "review Prototype"을 선택하고 "confirm"을 입력하면 업데이트가 완료됩니다.

1. 컴플라이언스 팀 구성원이 추가 설명이 필요한 제안으로 작업에 응답합니다.
2. The system sends a message to the product development team, requesting additional information or clarification.
2. 시스템은 추가 정보 또는 설명을 요청하는 메시지를 제품 개발 팀에 보냅니다.
3. The product development employee provides additional information or clarification.
3. 제품 개발 직원이 추가 정보 또는 설명을 제공합니다.
4. The system records the additional information or clarification provided by the compliance team and displays it to the product development team.
4. 컴플라이언스팀에서 제공한 추가 정보나 설명을 시스템에 기록하여 상품개발팀에 보여줍니다.

- A2, If compliance team members have disagreement among each other

(이해관계자 간에 의견 불일치가 있는 경우)

1. A compliance team employee responds to the task with a suggestion that conflicts with another employee's suggestion.

Test: After the product development team updates the prototype, send it again for another update. Until entering, the “confirm” compliance team can send the prototype back as many times as they want.

테스트: 제품 개발 팀이 프로토타입을 업데이트한 후 다른 업데이트를 위해 다시 보냅니다. 입장 전까지 "확인" 컴플라이언스 팀은 프로토타입을 원하는 만큼 여러 번 돌려보낼 수 있습니다.

1. 컴플라이언스 팀 직원이 다른 직원의 제안과 상충되는 제안으로 작업에 응답합니다.

2. The system sends a notification to the product development team, informing them of the conflict.

2. 시스템은 충돌을 알리는 알림을 제품 개발 팀에 보냅니다.

3. The product development team resolves the conflict by discussing the suggestions with the compliance team and arriving at a consensus.

3. 상품개발팀은 컴플라이언스팀과 제안사항을 논의하고 합의점에 도달하여 갈등을 해결합니다.

4. The system displays the resolved conflict and the consensus reached to the compliance team members.

4. 시스템은 해결된 갈등과 준법감시팀 구성원에게 도달한 합의를 표시합니다.

- E1, if Project Management Section having technical issues (프로젝트 관리 섹션에 기술적인 문제가 있는 경우)

1. The project management section is inaccessible due to a technical issue.

1. 기술적인 문제로 인해 프로젝트 관리 섹션에 액세스할 수 없습니다.

2. The system displays an error message to the product development team and the compliance team, informing them of the issue and suggesting an alternative method for providing input and printing " an unexpected error occurred, please try alternative ways to communicate".

2. 시스템은 제품 개발팀과 컴플라이언스 팀에 오류 메시지를 표시하여 문제를 알리고 입력 및 인쇄를 위한 대체 방법을 제안합니다. "예기치 않은 오류가 발생했습니다. 다른 통신 방법을 시도하십시오."

3. The issue persists and prevents the compliance team from providing their input within the given timeframe.

3. 문제가 지속되고 규정 준수 팀이 주어진 시간 내에 의견을 제공하지 못하게 합니다.

4. The system extends the timeframe or provides an alternative method for the compliance team to provide their input.

4. 시스템은 기간을 연장하거나 컴플라이언스 팀이 의견을 제공할 수 있는 대체 방법을 제공합니다.

- E2, when a compliance team member provides inappropriate or irrelevant input.
(이해관계자가 부적절하거나 관련 없는 의견을 제공하는 경우.)

1. A compliance team member provides inappropriate or irrelevant input.

1. 컴플라이언스 팀원이 부적절하거나 관련 없는 입력을 제공합니다.

2. The system removes the inappropriate or irrelevant input and sends a notification to the compliance team, informing them of the issue.

2. 시스템은 부적절하거나 관련 없는 입력을 제거하고 컴플라이언스팀에 문제를 알리는 알림을 보냅니다.

3. The compliance team member continues to provide inappropriate or irrelevant input.

3. 컴플라이언스 팀원이 계속해서 부적절하거나 관련 없는 정보를 제공합니다.

4. The system removes the compliance team employee's access to the task and sends a notification to the product development team, informing them of the issue.

4. 준법감시팀 직원의 업무접근권한을 제거하고, 상품개발팀으로 통보하여 문제를 알립니다.

Use Case Name : Assess risk and profitability / 위험 및 수익성을 평가한다

Test: Login as product development team employee after confirmation from compliance team.(prototype status is “confirmed” at this time.) Choose 4. Create a Risk Report.

Choose from Low/Medium/High options(optional) and write a report about what can be a problem. Prototype status changes to “standByFinal”. Log out and login again for Compliance Team Employees.

테스트: 컴플라이언스팀의 확인 후 제품개발팀 직원으로 로그인합니다.(현재 프로토타입 상태는 "확인됨"입니다.) 4. Create a Risk Report를 선택합니다. 낮음/중간/높음 옵션(선택 사항) 중에서

선택하고 무엇이 문제가 될 수 있는지에 대한 보고서를 작성합니다. 프로토타입 상태가 "standByFinal"로 변경됩니다. 준법감시팀 직원은 로그아웃 후 다시 로그인하세요.

Description : *This Use Case is to check the possible risks and ensure the profitability of the product. (이 Use Case는 발생할 수 있는 위험을 확인하고 제품의 수익성을 보장하기 위한 것입니다.)*

- **Basic Path**

1. The Compliance Team employee login with entering ID into a text box. (E1)
1. 상품개발팀 직원이 입력란에 아이디와 비밀번호를 입력하여 로그인한다. (E1)
2. System shows "Welcome + UserName".
2. 입력이 잘못된 경우 시스템에 "Welcome + UserName" 또는 "잘못된 ID/비밀번호"가 표시됩니다.
3. The compliance team accesses the task and reviews the requirements by pressing the "review Prototype" button.
3. 컴플라이언스 팀은 작업에 액세스하고 "검토" 버튼을 눌러 요구 사항을 검토합니다.
4. The compliance team can choose either "confirm" or "update" options.
 - If "confirm" is selected, prototype status changed to "confirmed" and then The Product Development Team can start creating a Risk Report without doing any update.
 - If "update" is selected, prototype status changes to "need update".
- 5. The Compliance Team selects option 2 (Provide Feedback) and writes feedback about what needs to be changed or improved and sends it back to The Product Development Team.
5. The product development employee receives the task notification and goes to "update prototype" section and makes necessary updates that were provided by The Compliance Team and sends it back to them. Prototype status once more goes back to "pending".
6. The compliance team approves the requirements if they are agreed by pressing the "confirm" button.
8. 컴플라이언스팀은 "확인" 버튼을 눌러 요구 사항에 동의하면 승인합니다.

9. The product development team assesses the risk associated with the new product or feature and creates a risk assessment report using the "create risk report" button and full fill necessary text boxes and press "confirm" button.

9. 제품 개발 팀은 새 제품 또는 기능과 관련된 위험을 평가하고 "위험 보고서 작성" 버튼을 사용하여 위험 평가 보고서를 작성하고 필요한 텍스트 상자를 모두 채우고 "확인" 버튼을 누릅니다.

- A1, If an update is necessary in risk requirements (위험요구사항 업데이트가 필요한 경우)

1.The product development employee creates a task for the prototype's risk features and sends it to the compliance team for review.

Test: Enter "provide Feedback" section and write your feedback in the Risk section.

Prototype status will be "need update".

테스트: "피드백 제공" 섹션을 입력하고 위험 섹션에 피드백을 작성합니다. 프로토타입 상태는 "업데이트 필요"입니다.

1. 제품 개발 직원이 프로토타입의 위험 기능에 대한 작업을 생성하고 검토를 위해 컴플라이언스 팀에 보냅니다.

2.The compliance team reviews the requirements and provides input and feedback to the product development team.

2. 컴플라이언스 팀은 요구 사항을 검토하고 제품 개발 팀에 입력 및 피드백을 제공합니다.

3. The product development team makes necessary adjustments to the requirements and updates the task status.

3. 제품 개발 팀은 요구 사항을 조정하고 작업 상태를 업데이트합니다.

4. The compliance team approves the requirements, and the product development team creates a risk assessment report.

4. 컴플라이언스 팀은 요구 사항을 승인하고 제품 개발 팀은 위험 평가 보고서를 작성합니다.

- A2, if product is not profitable (제품이 수익성이 없는 경우)

1. The compliance team identifies that product is not profitable and sends it back to the product development team for an update.

1. 컴플라이언스 팀은 제품이 수익성이 없음을 식별하고 업데이트를 위해 제품 개발 팀에 다시 보냅니다.

2. The product development team updates the product.

2. 제품 개발팀이 제품을 업데이트합니다.
 3. The compliance team reviews the additional information and confirms the product as profitable.
 3. 컴플라이언스팀은 추가 정보를 검토하고 제품이 수익성이 있는지 확인합니다.
- E1, In case of a miscommunication (잘못된 의사소통의 경우)
 1. The product development employee creates a task for the prototype's profitability features but forgets to send it to the compliance team for review.
 1. 제품 개발 직원이 프로토타입의 수익성 기능에 대한 작업을 생성했지만 검토를 위해 컴플라이언스 팀에 보내는 것을 잊었습니다.
 2. The compliance team does not receive the task notification, and the review process is delayed.
 2. 준법감시팀이 업무통보를 받지 못하고 심사가 지연된다.
 3. System prints an error "task is pending".
 3. 시스템이 "작업이 보류 중입니다"라는 오류를 인쇄합니다.
 - E2, in case task requirements are not fully filled. (작업 요구 사항이 완전히 채워지지 않은 경우.)
 1. Product development team fails to fully fill necessary fields for task requirements.
 1. 제품 개발 팀이 작업 요구 사항에 대한 필수 필드를 완전히 채우지 못했습니다.
 2. The compliance team rejects the task due to missing information.
 2. 컴플라이언스팀이 정보 누락으로 업무를 거부한 경우
 3. System prints "prototype is canceled".
 3. 시스템이 "시제품이 취소되었습니다"라고 인쇄합니다.

Use Case Name : Ensure compliance / 상품 출시를 인가한다.

Description : *This Use Case is to ensure compliance before launching the product. This is the last step of the prototype check process.*

Test: Login again as Compliance Team Employee and choose option "3. ensure Prototype" and choose the prototype you want to ensure and then choose 1. Approve

and enter a price. Prototype status will change to “approved” and it is ready to launch. Log out from the compliance team and login again as a product development team employee.

테스트: 컴플라이언스 팀 직원으로 다시 로그인하고 옵션 “3. 프로토타입 확인”을 선택하고 보장할 프로토타입을 선택한 다음 1을 선택합니다. 승인하고 가격을 입력합니다. 프로토타입 상태가 “승인됨”으로 변경되고 출시 준비가 됩니다. 컴플라이언스팀에서 로그아웃하고 상품개발팀 직원으로 다시 로그인합니다.

- Basic Path

1. The product development employee completes the risk report and sends the prototype to The Compliance Team for the final check and final confirmation. Prototype status changes to “standByFinalize” is this step.
1. 제품 개발 직원은 "새 작업" 버튼을 눌러 새 프로토타입의 적합성을 확인하기 위해 컴플라이언스 팀을 위한 새 작업을 만듭니다.
2. System notifies the compliance team about the new task by sending a notification that prints "new risk report created".
2. 시스템은 "새 작업 추가됨"을 인쇄하는 알림을 전송하여 새 작업에 대해 컴플라이언스 팀에 알립니다.
3. The compliance team reviews the risk report by pressing again with the “review prototype” option..
3. 컴플라이언스팀은 "열기" 버튼을 눌러 요구 사항을 검토합니다.
4. The compliance team checks the final prototype and its possible risks before approving the product.
4. 컴플라이언스팀에서 의견 및 피드백을 제공하고 "회신" 버튼을 누릅니다.(A1,A2)
5. The Compliance Team selects the ensure compliance section and either approves or cancels the all prototype.
 - If an approved option is selected, the system asks the Compliance Team to set up a price according to all information about the prototype and set up a payment term. Payment terms can be weekly, monthly, yearly or at once.
 - If the prototype has a High risk for not being profitable or for any other reason, the compliance team also can terminate the prototype in this stage by pressing the “ cancel prototype” button.

5. 제품 개발팀은 "열기/읽기" 버튼을 눌러 작업을 열고 컴플라이언스 팀의 피드백을 검토합니다.
 6. If the "approve" button is selected, The Product Development Team receives a notification about the prototype has been approved. (E1)
 6. Prototype status changes to "Approved" and only that time the prototype is ready to launch.
 7. 컴플라이언스 팀은 "승인" 버튼을 눌러 요구 사항이 준수되면 승인합니다.
 8. Product development team presses the "launch Prototype" button for launching the final prototype.
 8. 제품 개발 팀은 최종 프로토타입을 시작하기 위해 "시작" 버튼을 누릅니다.
- **A1, In case if compliance team has different suggestions**
 1. The compliance team provides feedback on the product requirements task.
 1. 컴플라이언스 팀은 제품 요구 사항 작업에 대한 피드백을 제공합니다.
 2. The product development team receives the message and makes adjustments to the requirements.
 2. 제품 개발 팀은 메시지를 수신하고 요구 사항을 조정합니다.
 3. The product development team updates the task status and assigns it back to the compliance team for review.
 3. 제품 개발 팀은 작업 상태를 업데이트하고 검토를 위해 규정 준수 팀에 다시 할당합니다.
 4. The compliance team receives the updated task and reviews the adjusted requirements.
 4. 컴플라이언스 팀은 업데이트된 작업을 수신하고 조정된 요구 사항을 검토합니다.
 - **A2, In case of an issue has been found by compliance team**
 1. The compliance team identifies a potential compliance issue with the product requirements.
 1. 규정 준수 팀이 제품 요구 사항에 대한 잠재적 규정 준수 문제를 식별합니다.
 2. The compliance team leaves comments on the task highlighting the issue and requesting adjustments.
 2. 컴플라이언스팀은 과제에 대한 의견을 남겨 문제를 강조하고 조정을 요청합니다.
 3. The product development team receives the comments and works with the compliance team to make necessary adjustments to the requirements.
 3. 제품 개발팀은 의견을 접수하고 규정 준수팀과 협력하여 요구사항에 필요한 조정을 합니다.

4. The product development team updates the task status and assigns it back to the compliance team for review.

4. 제품 개발 팀은 작업 상태를 업데이트하고 검토를 위해 규정 준수 팀에 다시 할당합니다.

- E1, In case of an issue that cannot be resolved

1. The compliance team identifies a major compliance issue with the product requirements that cannot be resolved through adjustments.

Test: After choosing 3. ensure Prototype , choose 2. Cancel and prototype will be canceled and deleted from the database.

테스트: 3. Prototype 확인 을 선택한 후 2. Cancel 을 선택하면 프로토타입이 취소되고 데이터베이스에서 삭제됩니다.

1. 컴플라이언스 팀은 조정을 통해 해결할 수 없는 제품 요구 사항의 주요 컴플라이언스 문제를 식별합니다.

2. The compliance team rejects the requirements and sends the task back to the product development team with a message explaining the reason for rejection.

2. 컴플라이언스 팀은 요구 사항을 거부하고 거부 이유를 설명하는 메시지와 함께 제품 개발 팀에 작업을 다시 보냅니다.

3. The product development team receives the task and works on revising the requirements.

3. 상품개발팀에서 과제를 접수받아 요구사항 수정 작업을 진행한다.

4. The product development team updates the task status and assigns it back to the compliance team for another review.

4. 제품 개발 팀은 작업 상태를 업데이트하고 다시 검토할 수 있도록 규정 준수 팀에 할당합니다.

- E2, In case if product team made a mistake

1. The product development team creates a task for the product requirements but forgets to assign it to the compliance team.

1. 제품 개발 팀이 제품 요구 사항에 대한 작업을 생성했지만 준수 팀에 할당하는 것을 잊었습니다.

2. The compliance team does not receive the task notification and no feedback is provided.

2. 컴플라이언스팀은 업무 알림을 받지 않으며 피드백을 제공하지 않습니다.

3. The product development team realizes the mistake and assigns the task to the compliance team.

3. 상품개발팀이 실수를 깨닫고 컴플라이언스팀에 업무를 맡긴다.
4. The compliance team receives the task and begins the review process.
4. 컴플라이언스 팀이 작업을 접수하고 검토 프로세스를 시작합니다.

- **E3, In case of a technical issue**

1. The compliance team experiences technical difficulties and is unable to access the project management system.
1. 컴플라이언스팀이 기술적인 어려움을 겪고 프로젝트 관리 시스템에 접근할 수 없습니다.
2. The compliance team is unable to provide feedback on the product requirements task.
2. 컴플라이언스 팀은 제품 요구 사항 작업에 대한 피드백을 제공할 수 없습니다.
3. The product development team follows up with the compliance team and learns about the technical difficulties.
3. 제품개발팀은 컴플라이언스팀과 후속 조치를 취하고 기술적인 어려움을 파악합니다.
4. The product development team works with the compliance team to find an alternative way for them to provide feedback, such as email or a phone call.
4. 제품 개발 팀은 규정 준수 팀과 협력하여 이메일이나 전화 통화와 같은 피드백을 제공할 수 있는 대체 방법을 찾습니다.

Usecase Name : 상품을 출시한다 Lunch a Product

Test: After getting approved from the Compliance Team, login as Product Development Team employee and choose 5. Lunch prototype. Enter “yes”. Prototype is launched as a new Insurance and prototype is deleted from the database.

테스트: 컴플라이언스팀의 승인을 받은 후 상품개발팀 직원으로 로그인하여 5. 런치 프로토타입을 선택합니다. "yes"를 입력합니다. 프로토타입이 새 보험으로 시작되고 프로토타입이 데이터베이스에서 삭제됩니다.

Description : *This is the Use Case where the prototype is ready and lunch as the product. It's included in the “Define Product Requirements” Use Case because it is the last step of the defining process. (이것은 프로토타입이 준비되고 실제 제품(보험상품)으로 출시되는 사례입니다. 정의 프로세스의 마지막 단계이기 때문에 "제품 요구 사항 정의" 사용 사례에 포함됩니다.) 유스케이스 다이어그램에서 이 유스케이스는 요구사항을 정의한다와 include로 이어져있기 때문에, 요구사항을 정의한다~상품 출시를 인가한다 까지 완료되어야 이 유스케이스로 이어질 수 있습니다.*

- Basic Path

1. The product development team checks the product details such as "productName","product ID", "product type" and "description" by choosing "show all prototypes" and entering prototype ID. (A1)

1. 제품 개발 팀은 "모든 프로토타입 보기"를 선택하고 프로토타입 ID를 입력함으로써 "제품명", "제품 ID", "제품 유형" 및 "설명"과 같은 제품 세부 정보를 확인합니다. (A1)

2. 제품 개발 팀이 "시제품 출시" 버튼을 클릭하여 제품을 출시할 준비가 된 것으로 표시합니다. 시스템은 작업 상태 변수를 확인합니다.

3. If the task status is "Approved", the system releases the product for launch and notifies the development team with a pop up prints "Successfully Launched". (A2)

3. 작업 상태가 "승인됨"인 경우 시스템은 출시를 위해 제품을 릴리스하고 "성공적으로 출시됨"이라는 팝업 메시지를 통해 개발 팀에 알립니다. (A2)

4.If the task status is not "Approved", the system shows the message "Only release a product with the approval of the compliance team. Please get permission and try again." and prompts the product development team to contact the compliance team for approval. (E1)

4. 작업 상태가 "승인됨"이 아닌 경우 "컴플라이언스 팀의 승인을 받은 제품만 릴리스하십시오. 권한을 받고 다시 시도하십시오."라는 메시지가 시스템에 표시됩니다. 승인을 위해 규정 준수 팀에 연락하도록 제품 개발 팀에 메시지를 표시합니다. (E1)

- A1, If the product development team does not complete all the product details. (제품 개발 팀이 모든 제품 세부 정보를 완료하지 않은 경우.)

1. If the product development team does not complete all the product details, the system shows an error message; "Please complete all required fields. Prototype creation couldn't be completed."

1. 제품 개발 팀이 모든 제품 세부 정보를 완료하지 않으면 시스템에 오류 메시지가 표시됩니다. "모든 필수 필드를 작성하십시오. 프로토타입 생성을 완료할 수 없습니다."

- A2, If the compliance team rejects the product (컴플라이언스팀이 제품을 거부하는 경우)

1. If the compliance team rejects the product, the system shows a message "Prototype canceled successfully!" and indicating the reason for rejection and prompts the product development team.
컴플라이언스 팀이 제품을 거부하면 시스템에 "프로토타입이 성공적으로 취소되었습니다!"라는 메시지가 표시됩니다. 거부 사유를 표시하고 제품 개발 팀에 메시지를 표시합니다.
- E1, If the task status is not "Approved" 작업 상태가 "준수 확인됨"이 아닌 경우
 1. If the task status is not "Approved" when the product development team tries to launch the product, the system shows the message "Only release a product with the approval of the compliance team. Please get permission and try again." and prompts the product development team to contact the compliance team for approval.
 1. 제품 개발팀에서 제품 출시를 시도할 때 작업 상태가 "승인됨"이 아닌 경우 "컴플라이언스 팀의 승인을 받은 제품만 출시하십시오. 허가를 받고 다시 시도하십시오."라는 메시지가 시스템에 표시됩니다. 승인을 위해 규정 준수 팀에 연락하도록 제품 개발 팀에 메시지를 표시합니다.

-상품 판매, 영업 및 마케팅 파트 (김민수)

Usecase Name : 보험 가입을 신청받는다.

Description : 영업팀 직원이 보험 시스템에 등록되어있는 보험 상품에 가입하고자 하는 고객의 가입 신청을 받는 액션이다.

테스트 : 프로그램 실행 후 -> booting iGO..통과한 후 (7초동안 입력 안하거나 아무 텍스트나 입력) -> 1번(고객 로그인) -> ics2024로 로그인 후 아래 시나리오대로 진행 (로그인 시 자사의 보험상품을 랜덤하게 광고하는 광고배너가 뜰 수 있는데, 이는 x. 닫기(한번 닫기)나 o. 접어두기(재로그인 전까지 닫기) 버튼을 클릭하고 진행하면 된다.)

- Basic Path
 1. 시스템은 초기화면(0. 보험 상품 확인, 1. 사고 접수, 2. 사고 조회, x. 로그아웃)을 보여준다.
 2. 고객은 보험 상품 확인 버튼을 누른다.
 3. 시스템은 보험 상품 메뉴 화면(보험 상품들, 1. 가입 신청하기, x. 뒤로가기)을 보여준다.
 4. 고객은 가입 신청하기 버튼을 누른다. (A1)

5. 시스템은 고객에게 개인정보 수집 및 이용 동의서를 보여준다.
6. 고객은 개인정보 수집 및 이용동의서에 동의란에 체크를 하고 다음 버튼을 누른다. (A2)
7. 시스템은 고객에게 "개인정보 수집 및 이용동의서에 동의해주셨습니다." 라는 메시지를 전송한다.
8. 고객은 확인 버튼을 누른다.
9. 시스템은 고객정보(이름, 주민등록번호, 전화번호, 계약하고자 하는 보험 Id) 제출 화면을 보여준다. (E1)

<고객 정보(ics2024) 입력창 예시>

이름: 이아파

주민등록번호: 9501192123424

전화번호: 01066339999

Id: si02

10. 고객은 고객정보(이름, 주민등록번호, 전화번호, 계약하고자 하는 보험 Id) 제출 버튼을 눌러 제출한다. (A3)

11. 시스템은 고객의 보험 가입 신청 내역을 영업팀 직원에게 전송한다. (E2)

- A1, 고객이 보험 가입을 원치 않는 경우

1. 고객은 뒤로가기 버튼을 누른다.
2. 시스템은 보험 상품 메뉴 화면을 송출한다.

- A2, 고객이 개인정보 수집 및 이용동의서에 동의를 하지 않을 경우

1. 고객은 개인정보 수집 및 이용동의서 동의서에 거절 버튼을 누른다.
2. 시스템은 "개인정보 수집 및 이용동의에 거절하셨습니다." 라는 메시지를 전송한다.
3. 고객은 확인 버튼을 누른다.
4. 시스템은 보험 상품 메뉴 화면을 송출한다. (E1)

- A3 고객 정보 입력에 오류가 있을 경우

1. 시스템은 고객에게 "고객 정보 입력에 오류가 있습니다. 고객 정보를 정확히 입력 후 제출해 주십시오."라는 메시지를 전송한다.
2. 고객은 확인 버튼을 누른다.
3. [Basic path] 9번부터 동일한 흐름으로 진행한다.

- E1, 화면이 7초 이상 나오지 않을 경우

1. 시스템은 고객에게 "시스템에 장애가 발생하였습니다. 고객센터에 직접 연락해 주십시오. 고객센터 번호 : 02-xxx-xxxx." 메시지를 전송한다.

- E2, 가입 신청 전송에 실패할 경우

1. 시스템은 고객에게 "가입 신청에 실패했습니다. 다시 시도해주시고. 지속적으로 실패할 경우 고객센터로 문의해주시기 바랍니다." 라는 메시지를 전송한다.

Usecase Name : 설문조사를 한다.

Description : 보험에 가입이 완료된 고객들에게 마케팅 팀에서 적절한 광고를 만들기 위해 설문조사를 하는 액션이다.

테스트 : 가입 신청하기 이후 설문조사 참여 (Y 입력) 이 후 시나리오대로 진행

- Basic Path

1. 시스템은 가입이 완료된 고객에게 "간단한 설문조사에 참여해 주시겠습니까?" 라는 메시지를 전송한다.
2. 고객은 "예" 버튼을 누른다. (A1)
3. 시스템은 고객에게 "나이를 선택해주시고.", "어떤 방식으로 자사 보험에 대해 알게 되었는지 선택해주시고.", "어떤 목적으로 자사 보험에 가입 하였는지 선택해주시고." 라는 질문과 각 질문별 답변 체크 박스가 보이는
(1. 20~29세, 2. 30~39세, 3. 40~49세, 4. 50~59세, 5. 60~69세, 6. 70세 이상 x. 설문조사 그만하기
1. 주변 지인의 추천, 2. 자사 직원의 추천, 3. 광고, 4. 기타 x. 설문조사 그만하기
1. 다른 사람들도 보험에 가입해서, 2. 혹시 모를 사고에 대비하고 싶어서, 3. 주변에서 보험에 가입하라고 조언해서, 4. 기타 x. 설문조사 그만하기)설문조사 화면을 송출한다. (E1)
4. 고객은 자신의 생각과 맞는 체크박스에 체크한다.
5. 고객은 제출 버튼을 누른다.(A2)(E2)
6. 시스템은 고객에게 "설문에 참여해주셔서 감사합니다" 라는 메시지를 전송한다.
- 7.고객은 확인 버튼을 누른다. → 해당 설문 결과는 마케팅 팀 직원으로 로그인 후 확인 가능 ex) 첫 로그인 화면(1. 회원 로그인, 2. 직원 로그인, x. 종료)에서 2(직원 로그인) 입력 후 me2023으로 로그인 → 마케팅 팀 메인화면에서 1(설문조사 결과 보기)입력 → 설문조사들의 결과 확인
- 8.시스템은 초기화면을 송출한다. (E3)

- A1, 고객이 설문조사에 참여하고 싶지 않을 경우

1. 고객은 "아니오" 버튼을 누른다.
2. 시스템은 고객에게 "보험에 가입해 주셔서 감사합니다." 메시지를 전송한다.
3. 고객은 확인 버튼을 누른다.

4. 시스템은 초기화면을 송출한다.
- A2, 고객의 변심으로 설문에 참여하고 싶지 않을 경우
 1. 고객은 "설문조사 그만하기" 버튼을 누른다.
 2. 보험사 시스템은 고객에게 "보험에 가입해 주셔서 감사합니다." 메시지를 전송한다.
 3. 고객은 확인 버튼을 누른다.
 4. 시스템은 초기화면을 송출한다.
- E1, 시스템이 설문조사 화면 송출에 실패했을 경우
 1. 시스템은 고객에게 "시스템에 장애가 발생하여 설문조사 참여에 실패하였습니다. 보험에 가입해 주셔서 감사합니다."라는 메시지를 전송한다.
- E2, 설문조사 제출에 실패할 경우
 1. 시스템은 고객에게 "시스템에 장애가 생겨 설문조사 제출에 실패했습니다. 보험에 가입해주셔서 감사합니다." 라는 메시지를 전송한다.
- E3, 화면이 7초 이상 나오지 않을 경우
 1. 시스템은 고객에게 "시스템에 장애가 발생하였습니다. 고객센터에 직접 연락해 주십시오. 고객센터 번호 : 02-xxx-xxxx." 메시지를 전송한다.

Usecase Name : 고객정보를 확보한다.

Description : 보험 가입을 신청한 고객이 전송한 고객의 정보들을 확인하고, 문제가 없는 고객 정보들을 UW팀에게 인수 심사 요청을 보내는 액션이다.

테스트 : 설문조사까지 완료된 이후 로그아웃하여 첫 로그인 화면에서 2(직원 로그인)입력 → se2023입력하여 로그인

- Basic Path
 1. 시스템은 고객에게 "보험 가입 신청이 완료되었습니다. 고객 정보 및 서류 검토 이후 SMS를 통해 결과를 전달하겠습니다." 라는 메시지를 전송한다.
 2. 고객은 확인 버튼을 누른다.
 3. 시스템은 고객에게 초기화면을 보여준다. (E1)
 4. 시스템은 영업팀 직원에게 영업팀 초기화면(1. 가입 신청 현황 확인, 2. 고객 정보 검색, x. 로그아웃)을 보여준다.
 5. 영업팀 직원은 가입 신청 현황 확인 버튼을 누른다.
 6. 시스템은 가입 신청 현황 화면(심사 상태가 "가입신청"인 보험 가입 신청 내역, 1. 심사 신청 보내기, 2. 심사 결과 확인, 3. 결과 전송하기, x. 뒤로가기)을 보여준다.

7. 영업팀 직원은 심사 신청 보내기 버튼을 누른다.
8. 시스템은 계약 ID, 피보험자 ID, 계약기간 입력화면을 보여준다.
9. 영업팀 직원은 정보를 입력한다. (A1)(A2)

<입력 예시>

Id: 2

피보험자 Id: ics2024

계약기간: 730

10. 시스템은 신청내역(고객 정보, 보험 정보)을 인수 심사 대상 리스트에 추가하고, "심사 요청이 완료되었습니다" 라는 메시지를 전송한다. (E2)

- A1, 영업팀이 없는 계약을 심사 신청 하였을 경우

1. 시스템은 영업팀 직원에게 "해당 ID의 계약이 없습니다." 라는 메시지를 전송한다.
2. 영업팀 직원은 확인 버튼을 누른다.
3. 시스템은 초기화면을 보여준다.

- A2, 영업팀 직원이 없는 고객정보를 입력하였을 경우

1. 시스템은 영업팀 직원에게 "해당하는 ID의 고객정보가 없습니다. ID를 다시 입력하여 주십시오." 라는 메시지를 전송한다.
2. [Basic path] 9번부터 동일한 흐름으로 진행한다.

- E1, 화면이 7초 이상 나오지 않을 경우

1. 시스템은 고객에게 "시스템에 장애가 발생하였습니다. 고객센터에 직접 연락해 주십시오. 고객센터 번호 : 02-xxx-xxxx." 메시지를 전송한다.

- E2, 인수 심사 대상 리스트 추가에 실패했을 경우

1. 시스템은 영업팀 직원에게 "시스템에 장애가 생겨 고개 정보 전달에 실패하였습니다. 시스템 관리자에게 문의해주십시오."라는 메시지를 전송한다.

Usecase Name : 보험 가입을 완료한다.

Description : 인수심사가 끝난 보험 가입 신청을 완료하고 이의 결과를 고객에게 전달하는 액션이다.

테스트 : UW팀의 인수심사를 한다 Use Case를 마친 이후 시나리오 - 인수심사를 완료한 후 다시 첫 로그인 화면에서 2(직원 로그인) 입력 후 se2023으로 로그인 한다. → 이후 시나리오대로 진행

- Basic Path

1. 시스템은 영업팀 직원에게 영업팀 초기화면(1. 가입 신청 현황 확인, 2. 고객 정보 검색, x. 로그아웃)을 보여준다. (E1)
2. 영업팀 직원은 가입 신청 현황 확인 버튼을 누른다.
3. 시스템은 가입 신청 현황 화면(심사 상태가 “가입신청”인 보험 가입 신청 내역, 1. 심사 신청 보내기, 2. 심사 결과 확인, 3. 결과 전송하기, x. 뒤로가기)을 보여준다.
4. 영업팀 직원은 심사 결과 확인 버튼을 누른다.
5. 시스템은 심사 상태 화면(심사 상태가 “승인”, “재심사 가능”, “재심사 거절”인 가입 신청 내역)을 보여준다. (E1)
6. 영업팀 직원은 가입 신청 현황 버튼을 누른 후 결과 전송하기 버튼을 누른다.
7. 시스템은 Id입력 화면을 보여준다.
8. 영업팀 직원은 결과를 전송하고자 하는 계약의 Id를 입력한다.
9. 시스템은 영업팀 직원에게 계약의 심사 상태가 “승인”일 경우 “해당 계약의 만료 날짜를 입력하세요.”라는 메시지를 전송한다.
10. 영업팀 직원은 계약 만료 날짜를 입력한다.

<입력 예시>

Id: 2

해당 계약의 만료 날짜를 입력하세요: (2023-06-11 기준 2025-06-08)

11. 시스템은 고객에게 “고객님의 “가입 신청한 보험 이름” 가입 신청의 결과가 도착했습니다. 결과: “승인”” 라는 메시지를 SMS를 통해 전송한다. (A1)(A2)(E2)
12. 시스템은 초기화면을 보여준다.

- A1, 심사 상태가 “재보험 거절”일 경우

1. 시스템은 고객에게 “고객님의 “가입 신청한 보험 이름” 가입 신청의 결과가 도착했습니다. 결과: “거절” 거절 사유: “거절 사유” 라는 메시지를 SMS를 통해 전송한다.

- A2, 심사 상태가 “재보험 가능”일 경우

1. 시스템은 고객에게 “고객님의 “가입 신청한 보험 이름” 가입 신청이 재심사에 들어가게 되어 추후에 결과를 알려드리겠습니다.” 라는 메시지를 SMS를 통해 전송한다.

- E1, 화면이 7초 이상 나오지 않을 경우

1. 시스템은 고객에게 “시스템에 장애가 발생하였습니다. 고객센터에 직접 연락해 주십시오. 고객센터 번호 : 02-xxx-xxxx.” 메시지를 전송한다.

- E2, 시스템이 SMS 전송을 실패할 경우

1. 시스템은 영업팀 직원에게 "SMS 전송에 실패하였습니다. 다시 시도해 주십시오. 실패가 지속되면 시스템 관리자에게 문의해 주십시오." 라는 메시지를 전송한다.

Usecase Name : 보험 상품을 추천한다

Description : 개발이 완료된 보험 상품을 해당 보험과 잘 맞는 고객들에게 판매하기 위해 관련된 고객들에게 보험을 추천하는 액션이다.

테스트 : 첫 로그인 화면에서 2(직원 로그인)입력 후 se2023으로 로그인 후 시나리오대로 진행

- Basic Path

1. 시스템은 영업팀 초기화면(1. 가입 신청 현황 확인, 2. 고객 정보 검색, x. 로그아웃)을 보여준다.
2. 영업팀 직원은 고객정보 검색 버튼을 누른다.
3. 시스템은 고객정보 검색 화면(1. 나이 검색, 2. 성별 검색, x. 뒤로가기)을 보여준다. (E1)
4. 영업팀은 원하는 조건(나이 또는 성별)을 입력하고 검색버튼을 누른다.

<입력 예시>

(1. 나이 검색 2. 성별 검색 x. 뒤로가기)화면 에서 1 또는 2 입력

1일 경우

→ 나이: 30

2일 경우

→ 성별: 남

5. 시스템은 입력된 조건에 일치하는 고객정보를 보여준다. (A1)(E2)
6. 영업팀은 조건에 부합하는 고객에게 전화를 통해 보험 상품을 추천한다.

- A1, 입력된 조건과 부합하는 고객이 없을 경우

1. 시스템은 영업팀 직원에게 "해당 나이의 고객이 없습니다." 또는 "해당 성별의 고객이 없습니다." 라는 메시지를 전송한다.
2. [Basic path] 3번부터 동일한 흐름으로 진행한다.

- E1, 화면이 7초 이상 나오지 않을 경우

1. 시스템은 고객에게 "시스템에 장애가 발생하였습니다. 고객센터에 직접 연락해 주십시오. 고객센터 번호 : 02-xxx-xxxx." 메시지를 전송한다.

- E2, 고객정보 검색에 실패하였을 경우

1. 시스템은 영업팀에게 "시스템에 장애가 발생하여 검색에 실패하였습니다. 시스템 관리자에게 문의해주십시오." 메시지를 전송한다.

Usecase Name : 보험 상품 광고를 등록한다.

Description : 마케팅 팀이 고객들에게 새로운 보험 상품을 광고하여 보험 가입을 유도하는 액션이다.

테스트 : 첫 로그인 화면에서 2(직원 로그인)입력 후 me2023으로 로그인 후 시나리오대로 진행

- Basic Path

1. 시스템은 마케팅 팀 초기화면(1. 설문조사 결과 보기, 2. 광고 등록하기, x. 로그아웃)을 보여준다.
2. 마케팅 팀은 광고 등록하기 버튼을 누른다.
3. 시스템은 광고 등록 화면("광고하고자 하는 상품의 이름을 적어주세요.", "광고하고자 하는 상품의 설명을 적어주세요." x. 뒤로가기)을 보여준다
4. 마케팅 팀은 상품의 이름과 설명을 등록한다. (A1)
<입력 예시>
상품 이름: bb 자동차 보험
상품 설명: 마음 놓고 운전할 수 있도록 도와주는 자동차 보험
5. 시스템은 마케팅 팀에게 "광고 등록이 완료되었습니다." 라는 메시지를 전송한다. (E1)
6. 시스템은 보험 상품 광고를 초기화면 배너에 등록한다. (A2)(E2)(E3)

- A1, 광고 등록을 하지 않을 경우

1. 마케팅 팀 직원은 등록 취소 버튼을 누른다.
2. 시스템은 마케팅 팀 직원에게 "광고 등록을 취소합니다." 라는 메시지를 전송한다.

- A2, 고객이 광고를 그만 보고 싶은 경우

1. 고객은 접어두기 버튼을 누른다.
2. 시스템은 해당 고객이 재로그인 하기 전까지 광고를 없애준다.

- E1, 광고 등록에 실패했을 경우

1. 시스템은 마케팅 팀 직원에게 "광고 등록에 실패했습니다. 시스템 관리자에게 문의해주시오"라는 메시지를 전송한다.

- E2, 광고가 배너에 등록되지 않은 경우

1. 시스템은 마케팅 팀 직원에게 "초기화면 배너에 광고 등록을 실패했습니다. 시스템 관리자에게 문의해주시오"라는 메시지를 전송한다.

-계약 관리, 보험료 납부 파트 (김나연)

Usecase Name : 보험 계약을 관리한다.

Description : 계약관리팀이 보험료 미납, 계약 만기임박, 보험 부활 대상 고객을 조회하고, 해당 사실 혹은 계약 변동사항을 고객에게 안내하는 액션이다. 만일 보험 부활이라면 대상 고객 조회 후, extend로 연결된 '보험 부활을 심사한다' UseCase로 이동한다.

- Basic Path

1. 계약관리팀은 보험사 시스템 어플리케이션 초기 화면에서 로그인한다.
2. 시스템은 "1. 보험료 미납 고객 조회 2. 만기 임박 고객 조회 3. 부활 신청 고객 조회 x. 로그아웃" 화면을 보여준다.
3. 계약관리팀은 1을 입력한다. (A1, A2)
4. 시스템은 미납 고객에 대한 정보(이름, 전화번호, 미납 횟수, 월 보험료, 보험상품명)를 출력한다. (E2)
5. 계약관리팀은 미납 안내할 고객 이름을 입력한다.
6. 시스템은 해당 고객의 미납횟수가 3회 미만일 경우, 계약관리팀은 미납 고객에게 시스템을 통해 '보험료 미납 안내 전화를 하시겠습니까? (y/n)' 메시지를 보여준다. (A3)
7. 계약관리팀은 y를 입력한다.
8. 시스템은 해당 고객에게 전화를 발신한다.
9. 계약관리팀은 고객에게 미납 안내 사항을 모두 전달하면 x를 입력해 통화를 종료한다.

- A1, 2(만기 임박 고객 조회)를 입력한 경우

1. 시스템은 계약 만기 임박(만기일이 30일 이하로 남은) 고객에 대한 정보(이름, 전화번호, 만기일, 보험상품명)를 출력한다.
2. 계약관리팀은 계약 만기 임박 고객에게 시스템을 통해 보험 만기 안내 전화를 한다.
3. 고객은 보험사의 계약 만기 안내 전화를 수신한다. (E1)
4. 고객은 재계약 여부를 계약관리팀 담당자에게 전달한다.
5. 계약관리팀은 고객의 재계약 여부에 따라 시스템에 변경된 만기일을 입력한다.

- A2, 3(부활 신청 고객 조회)을 입력한 경우

1. 시스템은 보험계약 부활을 신청한 고객에 대한 정보(이름, 전화번호, 보험상품명, 실효일, 총 미납료, 미납료 납입 여부)를 출력한다.
2. (UseCase '보험 부활을 심사한다'로 이동)

- A3, 미납횟수가 3회 이상일 경우

1. 시스템은 미납횟수가 3회 이상일 경우, '실효보험 처리를 하시겠습니까? (y/n)' 메뉴를 보여준다.
2. 계약관리팀 직원은 y를 입력한다.
3. 시스템은 고객에게 보험 계약이 실효됨을 알리는 문자(“

[XX보험 OOOO보험 계약 실효 안내]

OOO 고객님의 가입하신 OOOO보험이 실효됨을 안내드립니다.

보험 부활을 원하시면 iGO보험 사이트에 접속하셔서 부활 신청을 접수해주시거나, 아래 번호로 문의전화 바랍니다.

-보험명 : OOOO보험

-문의전화 : 02-XXXX-XXXX

”)를 전송한다.

- E1, 고객이 계약 만기 안내 전화를 수신하지 않는 경우

1. 계약관리팀은 고객에게 보험 계약 만기 임박 안내 사항(“

[XX보험 OOOO보험 계약 만기 임박 안내]

OOO 고객님의 가입하신 OOOO보험의 만기일이 30일 남아 안내드립니다.

계약 연장을 원하시면 재계약 여부를 전화 혹은 문자로 전달해주시기 바랍니다.

-보험명 : OOOO보험

-보장기간 : 20XX.XX.XX ~ 20XX.XX.XX (N년)

-문의전화 : 02-XXXX-XXXX

”) 문자를 전송한다.

Usecase Name : 보험 부활을 심사한다.

Description : 계약관리팀이 보험 부활 심사 현황을 조회하고 심사 결과를 입력하는 액션이다. 부활 심사 결과가 확정되면 시스템에 반영하고 고객에게 심사 결과를 안내한다.

- Basic Path

1. 계약관리팀 직원은 보험 부활을 신청한 고객 목록 중 심사할 고객 이름을 클릭한다.
2. 시스템은 해당 고객의 심사 상태, 고객과 부활 대상 보험의 상세정보(고객 이름, 성별, 나이, 전화번호, 요율, 보험명, 가입일, 실효일, 약관)를 보여준다.
3. 계약관리팀 직원은 '부활 확정' 버튼을 클릭한다. (A1)
4. 시스템은 '보험 부활을 확정하시겠습니까?' 라는 모달창을 띄운다.
5. 계약관리팀 직원은 '확인' 버튼을 누른다. (A2)
6. 시스템은 해당 고객의 심사 상태를 '부활 확정'으로 변경하고, 고객에게 부활 확정 안내 사항(“

[XX보험 0000보험 부활 심사 결과 안내]

000 고객님의 신청하신 0000보험의 부활 심사 결과를 안내드립니다.

심사 결과, 보험 부활이 승인되었습니다.

N월 NN일부터 보험료 납입 예정이니 아래 사항 참고바랍니다.

-보험명 : 0000보험

-납입일 : 매달 NN일

-보험료 : XX,XXX원

-보장기간 : 20XX.XX.XX ~ 20XX.XX.XX (N년)

-보장내역 : {각 고객별 가입 보장 내역 작성}

-문의전화 : 02-XXXX-XXXX

)을 문자로 전송한다.

- A1, '부활 거절' 버튼을 클릭한 경우

1. 계약관리팀 직원이 '부활 거절' 버튼을 누른다.

2. 시스템은 '보험 부활을 거절하시겠습니까?' 라는 모달창을 띄운다.

3. 계약관리팀 직원이 '확인' 버튼을 클릭한다. (A2)

4. 시스템은 해당 고객의 심사 상태를 '부활 거절'으로 변경하고, 고객에게 부활 거절 안내 사항(“

[XX보험 0000보험 부활 심사 결과 안내]

000 고객님의 신청하신 0000보험의 부활 심사 결과를 안내드립니다.

심사 결과, 보험 부활이 거절되었습니다.

-보험명 : 0000보험

-문의전화 : 02-XXXX-XXXX

)을 문자로 전송한다.

- A2, 모달창에서 '취소' 버튼을 누를 경우

1. 시스템은 모달창을 닫고, 해당 고객의 심사 상태, 고객과 부활 대상 보험의 상세정보(고객 이름, 성별, 나이, 직업, 요율, 전화번호, 보험명, 가입일, 실효일, 약관)를 보여준다.

- E1, 화면이 7초 이상 나오지 않을 경우

1. 시스템은 "화면을 불러오지 못하고 있습니다. 같은 현상이 반복되면 고객센터에 신고해 주십시오."라는 메시지를 화면에 보여준다.

2. 계약관리팀 직원은 확인 버튼을 클릭한다.

3. 시스템은 어플리케이션 초기화면을 보여준다.

Usecase Name : 월 보험료를 납입받는다.

Description : 고객이 보험료 납입 내역을 조회하고, 보험료를 납입하기 위한 결제 프로세스가 진행되는 액션이다.

- Basic Path

Precondition : 고객은 보험 상품에 가입을 한 상태

1. 고객은 보험사 시스템 어플리케이션 초기 화면에서 마이페이지 메뉴를 클릭한다.
2. 시스템은 마이페이지 화면에서 고객이 가입한 보험 내역(보험명, 가입일, 만기일, 납입일, 월 보험료, 미납횟수, 미납료)과 '납입하기' 버튼을 보여준다. (E3)
3. 고객은 '납입하기' 버튼을 클릭한다.
4. 시스템은 보험료 납입 결제창(결제 금액, 결제 수단, 결제 버튼)을 띄운다. (E1)
5. 고객은 원하는 결제 수단을 선택한 후, '결제' 버튼을 누른다. (A1)
6. 시스템은 성공적으로 결제를 마쳤을 경우, "정상적으로 처리되었습니다." 라는 모달창을 띄운다. (E2)
7. 고객은 '확인' 버튼을 누른다.
8. 시스템은 결제창을 종료하고 마이페이지 화면으로 돌아가 고객이 가입한 보험내역(보험명, 가입일, 만기일, 납입일, 월 보험료, 미납료, 납부방식)을 보여준다.

- A1, 결제창 닫기(X) 버튼을 클릭한 경우

1. 시스템은 결제창을 종료하고 마이페이지를 보여준다.

- E1, 결제창이 7초 이상 나오지 않을 경우

1. 시스템은 결제창을 종료하고 "오류가 발생했습니다. 잠시 후에 다시 시도해주시시오"라는 메시지를 화면에 보여준다.
2. 계약관리팀 직원은 확인 버튼을 클릭한다.
3. 시스템은 어플리케이션 초기화면을 보여준다.

- E2, 보험료 결제에 실패했을 경우

1. 시스템은 "보험료 납입에 실패하였습니다. 잠시 후에 다시 시도해주시시오."라는 모달창을 띄운다.
2. 고객은 '확인' 버튼을 누른다.
3. 시스템은 결제창을 종료하고 마이페이지를 보여준다.

- E3, 화면이 7초 이상 나오지 않을 경우

1. 시스템은 "화면을 불러오지 못하고 있습니다. 같은 현상이 반복되면 고객센터에 신고해 주십시오."라는 메시지를 화면에 보여준다.

2. 고객은 확인 버튼을 클릭한다.
3. 시스템은 어플리케이션 초기화면을 보여준다.

-인수심사(U/W), 재보험 파트 (유병규)

Usecase Name : 인수 심사를 한다.

Description : U/W 부서 직원이 보험에 대해 인수 심사를 진행하기 위한 액션이다.

테스트 : 프로그램 실행 후 -> 'Booting iGO...'에서 초기 'DBFuntion'을 통해 DB초기화 진행하였다면
통과 (7초동안 입력 안하거나 아무 텍스트나 입력)->2번(직원 로그인) -> uw01으로 로그인 후 아래
시나리오대로 진행

- Basic Path

1. U/W 부서 직원은 시스템 어플리케이션 초기화면에서 인수심사 메뉴를 클릭한다.
2. 시스템은 인수심사 초기화면(인수 심사 대기 리스트(계약 ID, 보험 ID, 담당 영업팀 직원 ID, 보험계약자 ID, 피보험자 ID, 인수 심사 상태)와 메시지("인수 심사를 진행할 계약 ID를 선택해주세요."))를 보여준다.(E1)
3. U/W부서 직원은 심사할 화면에 보이는 리스트 중에서 인수 심사할 계약 ID를 입력한다.
4. 시스템은 인수심사 화면(인수 심사 계약 정보(피보험자 정보(고객 ID, 이름, 주민등록번호), 가입 보험 정보(보험 ID, 이름, 지급 금액), 계약 내용(계약 금액, 보험료 납부 주기, 계약 기간))을 보여준다.
5. U/W 부서 직원은 손해율 측정 버튼을 클릭한다.
6. 시스템은 '손해율을 분석한다' Usecase를 통해 손해율 분석 메시지("손해율 분석이 완료되었습니다."예상 고객 납부금액, 예상 지급금액, 손해율)와 버튼("1.인수심사 2.인수심사 거절")을 보여준다.(E2)
7. U/W 부서 직원은 인수심사 버튼을 클릭한다.(A2)
8. 시스템은 금융감독원에게 보험 계약 내용을 SMS로 전송하고 승인, 거절 버튼을 보여준다.
9. 금융감독원은 승인 버튼을 누른다.(A3)
10. 시스템은 해당 계약을 '승인'으로 표시하고 심사 결과 메시지("인수 심사가 완료되었습니다." 심사 대상자 (이름, 주민등록번호), 가입 보험 이름, 가입 기간, 가입 유무)와 버튼("1.확인")을 U/W직원에게 보여준다.
11. U/W 부서 직원은 확인 버튼을 누른다.
12. 시스템은 초기화면(UW팀 메뉴)를 보여준다.

- **A1, 재심사 버튼을 클릭했을 경우**

1. 시스템은 재심사 사유란을 보여준다.
2. U/W 직원은 재심사 사유란을 작성한다
3. 시스템은 재심사 버튼을 보여준다.
4. U/W 직원은 재심사 버튼을 클릭한다.
5. 시스템은 재심사 사유와 보험 계약 내용을 금융감독원에게 SMS로 전송한다.
6. 금융감독원은 승인 버튼을 누른다.(A4)
7. 시스템은 해당 계약을 '승인'으로 표시하고 심사 결과 메시지("인수 재심사가 완료되었습니다."
재심사 대상자 (이름, 주민등록번호), 가입 보험 이름, 가입 기간, 가입 유무)와 버튼("1.확인")을
출력한다.
8. U/W 부서 직원은 확인 버튼을 누른다.
9. 시스템은 초기화면(UW팀 메뉴)를 보여준다.

- **A2, 인수 심사를 거절할 경우**

1. U/W 부서 직원은 인수 심사 거절 버튼을 누른다.
2. 시스템은 '거절 사유란'을 보여준다.
3. U/W 부서 직원 거절 사유를 입력하고 확인 버튼을 누른다.
4. 시스템은 해당 계약을 '거절'로 표시하고 심사 결과 메시지("인수 심사를 거절하였습니다."
거절사유)와 버튼("1.확인")을 출력한다.
5. U/W 부서 직원은 확인 버튼을 누른다.
6. 시스템은 초기화면(UW팀 메뉴)를 보여준다.

- **A3, 금융감독원이 거절 버튼을 눌렀을 경우**

1. 시스템은 거절 사유 입력란을 보여준다.
2. 금융감독원은 거절 사유를 입력하고 전송하기 버튼을 누른다.
3. 시스템은 U/W 부서 직원에게 심사 결과 메시지("인수 심사가 거절되었습니다." 심사
대상자(이름, 주민등록번호), 가입 보험 이름, 가입 기간, 가입 유무, 거절 사유)와 버튼("1.확인
2.재심사")을 보여준다.
4. U/W 부서 직원은 확인 버튼을 누른다.(A1)
5. 시스템은 초기화면(UW팀 메뉴)를 보여준다.

- **A4, 금융감독원이 재심사 거절 버튼을 눌렀을 경우**

1. 시스템은 거절 사유 입력란을 보여준다.
2. 금융감독원은 거절 사유를 입력하고 전송하기 버튼을 누른다.

3. 시스템은 해당 계약을 '재심사 거절'로 표시하고 U/W 부서 직원에게 심사 결과 메시지("인수 재심사가 거절되었습니다." 재재심사 대상자(이름, 주민등록번호), 가입 보험 이름, 가입 기간, 가입 유무, 거절 사유)와 버튼("1.확인")을 보여준다.

4. U/W 부서 직원은 확인 버튼을 누른다.

5. 시스템은 인수심사 초기화면(UW팀 메뉴)을 보여준다.

- E1, 화면이 7초 이상 나오지 않을 경우

1. 시스템은 "화면을 불러오지 못하고 있습니다. 같은 현상이 반복되면 고객센터에 신고해 주십시오."라는 메시지를 화면에 보여준다.

2. U/W 부서 직원은 확인 버튼을 클릭한다.

3. 시스템은 어플리케이션 초기화면을 보여준다.

- E2, 손해율 측정에 실패했을 경우

1. 시스템은 손해율 측정 실패 메시지("손해율 측정에 실패하였습니다. 다시 시도해주세요.")를 보여준다.

2. U/W 직원은 메시지를 수신한다.

3. 시스템은 UW팀 초기화면을 보여준다.

Usecase Name : 재보험 등록을 한다.

Description : 재보험 회사와 재보험 계약을 진행하기 위한 액션이다.

테스트 : 2번(직원 로그인) -> uw01으로 로그인 후 아래 시나리오대로 진행

- Basic Path

1. U/W 부서 직원은 시스템 어플리케이션 초기화면에서 재보험 등록 메뉴를 클릭한다.

2. 시스템은 재보험 등록 초기화면(재보험 대상 계약 ID, 재보험 회사 이름, 재보험 회사 담당자 이름, 재보험 회사 담당자 연락처, 재보험 계약 기간, 재보험 계약금, 재보험 계약 비율)을 보여준다.(E1)

3. U/W 부서 직원은 내용을 입력하고 손해율 분석 버튼을 누른다.

4. 시스템은 '손해율을 분석한다' Usecase를 통해 손해율을 보여준다.(E2)

5. U/W 직원은 재보험 등록 요청 진행하기 버튼을 클릭한다.(A1)

6. 시스템은 재보험 회사 담당자에게 재보험 등록 내용을 전송한다.(E3)

7. 재보험 회사 담당자는 계약 승인 버튼을 누른다.(A2)

8. 시스템은 재보험 등록이 완료되었다는 메시지("재보험 등록이 완료되었습니다." 재보험 대상 계약 ID, 재보험 회사 이름, 재보험 회사 담당자 이름, 재보험 회사 담당자 연락처, 재보험 계약 기간, 재보험 계약금, 재보험 계약 비율))를 U/W 부서 직원에게 제공한다.

9. U/W 부서 직원은 메시지를 수신한다.

- **A1, 재보험 등록 요청 그만두기 버튼을 클릭할 경우**

1. 시스템은 확인 메시지("재보험 등록 요청을 취소하시겠습니까?")를 보여준다.

2. U/W 직원은 확인 버튼을 클릭한다.

3. 시스템은 취소 메시지("재보험 등록 요청이 취소되었습니다.")를 보여주고 초기화면(UW팀 메뉴)을 보여준다.

- **A2, 요청 거절 버튼을 클릭할 경우**

1. 시스템은 거절 사유 입력란을 보여준다.

2. 재보험 회사 담당자는 거절 사유를 입력하고 확인버튼을 클릭한다.

3. 시스템은 U/W 직원에게 메시지("재보험 신청이 거절되었습니다." 거절사유)를 보여준다.

4. U/W 직원은 메시지를 수신한다.

- **E1, 화면이 7초 이상 나오지 않을 경우**

1. 시스템은 "화면을 불러오지 못하고 있습니다. 같은 현상이 반복되면 고객센터에 신고해 주십시오."라는 메시지를 화면에 보여준다.

2. U/W 부서 직원은 확인 버튼을 클릭한다.

3. 시스템은 어플리케이션 초기화면을 보여준다.

- **E2, 손해율 측정에 실패했을 경우**

1. 시스템은 손해율 측정 실패 메시지("손해율 측정에 실패하였습니다. 다시 시도해주세요.")를 보여준다.

2. U/W 직원은 메시지를 수신한다.

3. 시스템은 재보험 등록 초기화면(U/W직원이 입력한 내용을 유지한 상태의 화면)을 보여준다.

- **E3, 재보험 회사 담당자가 10분동안 응답이 없을 경우**

1. 시스템은 재보험 회사 담당자에게 재요청을 보낸다.

2. 시스템은 U/W 직원에게 안내 메시지("현재 재보험 등록 요청에 대한 응답이 없어 재요청 하였습니다.")를 보여준다.

Usecase Name : 손해율을 분석한다.

Description : 보험 인수, 재보험 등록에 대해 손해율을 분석하기 위한 액션이다.

테스트 : 2번(직원 로그인) -> uw01으로 로그인 후 아래 시나리오대로 진행

- Basic Path

1. U/W 부서 직원은 시스템 어플리케이션 초기 화면에서 손해율 분석 메뉴를 클릭한다.
2. 시스템은 손해율 분석 초기화면("1. 보험 인수 손해율 계산, 2. 재보험 등록 손해율 계산")을 보여준다.(E1)
3. U/W 부서 직원은 손해율 분석할 '보험 인수 손해율 계산'을 선택한다.(A1)
4. 시스템은 메시지("손해율을 계산할 계약 ID를 입력해주세요.")를 출력한다.
5. U/W 부서 직원은 계약 ID를 입력한다.
6. 시스템은 '고객 정보를 요청한다' Usecase를 통해 계약에 해당하는 고객 정보(이름, 주민번호, 주소, 재산, 신용정보)와 보험 정보(보험 상품 종류, 상품 이름, 지급 금액, 계약 금액, 계약 기간)를 출력한다.(A2)
7. U/W 부서 직원은 손해율 분석 버튼을 클릭한다.(E2)
8. 시스템은 손해율 분석이 완료되었다는 메시지("손해율 분석이 완료되었습니다." 예상 고객 납부금액, 예상 지급금액, 손해율)를 U/W 부서 직원에게 제공한다.

- A1, 재보험 등록을 위한 손해율 분석일 경우

1. 시스템은 메시지("손해율을 계산할 재보험 ID를 입력해주세요.")를 출력한다.
2. U/W 부서 직원은 재보험 ID를 입력한다.
3. 시스템은 U/W 재보험 회사 정보(재보험 회사 이름, 재보험 회사 담당자 이름, 재보험 회사 담당자 연락처)와 계약 정보(계약 금액, 계약 조건, 재보험 대상 계약ID)을 출력한다.
4. U/W 부서 직원은 손해율 분석 버튼을 클릭한다.
5. 시스템은 손해율 분석이 완료되었다는 메시지("손해율 분석이 완료되었습니다." 예상 고객 납부금액, 예상 지급금액, 예상 손해율)을 U/W 부서 직원에게 제공한다.

- A2, 고객정보 요청이 거절 되었을 경우

1. 시스템은 U/W 직원에게 메시지("고객 정보 요청이 거절되었습니다." 거절사유)를 보여준다.
2. U/W 직원은 메시지를 수신한다.
3. 시스템은 손해율 분석 메뉴 초기화면을 보여준다.

- E1, 화면이 7초 이상 나오지 않은 경우

1. 시스템은 "화면을 불러오지 못하고 있습니다. 같은 현상이 반복되면 고객센터에 신고해 주십시오."라는 메시지를 화면에 보여준다.
2. U/W 부서 직원은 확인 버튼을 클릭한다.

3. 시스템은 어플리케이션 초기화면을 보여준다.

- E2, 손해율 측정에 실패했을 경우

1. 시스템은 손해율 측정 실패 메시지("손해율 측정에 실패하였습니다. 다시 시도해주세요.")를 보여준다.

2. U/W 직원은 메시지를 수신한다.

3. 시스템은 어플리케이션 초기화면을 보여준다.

Usecase Name : 고객 정보를 요청한다.

Description : 인수 심사에 필요한 고객 정보를 은행담당원에게 요청하여 정보를 받아오기 위한 액션이다.

테스트 : 2번(직원 로그인) -> uw01으로 로그인 후 아래 시나리오대로 진행

- Basic Path

1. U/W 부서 직원은 시스템 어플리케이션 초기화면에서 '고객 정보 요청' 메뉴를 클릭한다.

2. 보험사 시스템은 '고객 정보 요청' 초기 화면(고객 이름, 고객 주민등록번호, 요청 사유)을 보여준다.(E1)

3. U/W 부서 직원은 고객 정보를 입력하고 요청하기 버튼을 클릭한다.

4. 시스템은 은행담당원에게 요청할 고객의 정보를 SMS로 전송한다.(E2)

5. 은행담당원은 요청 수락 버튼을 클릭한다.(A1)

6. 시스템은 선택 메시지("1. 고객 정보 입력하기, 2. 고객 정보가 존재하지 않음")를 보여준다.

7. 은행담당원은 고객 정보 입력하기를 선택한다.(A2)

8. 시스템은 고객 정보 입력 화면(이름, 나이, 주소, 재산, 신용정보(신용점수, 조회내역, 변동내역, 월 카드 사용금액, 대출내역, 연체내역, 연대보증 내역))을 보여준다.

9. 은행담당원은 해당 고객의 정보(이름, 나이, 주소, 재산, 신용정보(신용점수, 조회내역, 변동내역, 월 카드 사용금액, 대출내역, 연체내역, 연대보증 내역))를 입력하고 전송 버튼을 클릭한다.(E3)

10. 시스템은 U/W 부서 직원에게 고객 정보 조회 완료 메시지를 ("고객 정보가 조회 되었습니다."

고객: 이름, 주민등록번호, 고객의 정보: 이름, 나이, 주소, 재산, 신용정보(신용점수, 조회내역, 변동내역, 월 카드 사용금액, 대출내역, 연체내역, 연대보증 내역)) 제공한다.

11. U/W 부서 직원은 고객 정보를 수신하여 확인한다.

- A1, 요청 거절 버튼을 클릭할 경우

1. 시스템은 거절 사유 입력란을 보여준다.

2. 은행담당원은 거절 사유를 입력하고 전송하기 버튼을 클릭한다.

3. 시스템은 U/W 직원에게 메시지("고객 정보 요청이 거절되었습니다." 거절사유)를 보여준다.
 4. U/W 직원은 메시지를 수신한다.
- **A2, 고객 정보(이름, 주민등록번호)가 존재하지 않을 경우**
 1. 은행담당원은 고객 정보가 존재하지 않음을 선택한다.
 2. 시스템은 거절 사유 입력란을 보여준다.
 3. 은행담당원은 거절 사유를 입력하고 전송하기 버튼을 클릭한다.
 4. 시스템은 U/W 직원에게 메시지("고객 정보 요청이 거절되었습니다." 거절사유)를 보여준다.
 5. U/W 직원은 메시지를 수신한다.
 - **E1, 화면이 7초 이상 나오지 않을 경우**
 1. 시스템은 "화면을 불러오지 못하고 있습니다. 같은 현상이 반복되면 고객센터에 신고해 주십시오."라는 메시지를 화면에 보여준다.
 2. U/W 부서 직원은 확인 버튼을 클릭한다.
 3. 시스템은 어플리케이션 초기화면을 보여준다.
 - **E2, 은행담당원이 10분동안 응답이 없을 경우**
 1. 시스템은 은행담당원에게 재요청을 보낸다.
 2. 시스템은 U/W 직원에게 안내 메시지("현재 고객 정보 요청에 대한 응답이 없어 재요청 하였습니다.")를 보여준다.
 - **E3, 고객 정보 전송에 실패하였을 경우**
 1. 시스템은 은행담당원에게 전송 실패 메시지("고객 정보 전송에 실패하였습니다. 잠시 후 다시 시도해주세요.")를 보여준다.
 2. 은행담당원은 확인 버튼을 클릭한다.
 3. 시스템은 고객 정보 입력 화면(은행담당원이 입력한 내용을 유지한 화면)을 보여준다.

-사고 접수, 보상 처리 파트 (김건우)

Usecase Name : 사고를 접수한다.

Description : 보험이 적용되는 피보험자에게 사고 등의 발생으로 보상을 받아야 할 때 고객은 이를 보험사에 알리고, 보험사의 사고 접수 직원이 접수를 처리하는 액션이다. 여기서 말하는 사고접수란, 손해보험에서 발생한 자동차 사고등도 있겠지만, 보다 넓은 의미로 생명보험에서 질병에 걸렸다거나의 경우도 모두 사고(사건)의 발생으로 보고, 이를 보상받기 위해 접수하는것으로 여긴다.

테스트 : 프로그램 실행 후 -> booting iGO..통과한 후 (7초동안 입력 안하거나 아무 텍스트나 입력) -> 1번(고객 로그인) -> ics2023으로 로그인 후 아래 시나리오대로 진행 (로그인 시 자사의 보험상품을 랜덤하게 광고하는 광고배너가 뜰 수 있는데, 이는 닫거나 접어두기 버튼을 클릭하고 진행하면 된다.)

- Basic Path

1. 피보험자는 보험 시스템 어플리케이션 초기화면(사고 접수, 사고 조회 메뉴 존재)에서 사고 접수 메뉴에 들어간다. (E1)

2. 시스템은 1.일반 접수하기 2.긴급 접수하기 3.뒤로가기 버튼들을 보여준다.

-> 긴급 접수하기 클릭 시의 상황은 긴급 접수한다 유스케이스 시나리오로

2-1. 피보험자는 일반 접수하기 버튼을 누른다.

3. 시스템은 아래의 입력 폼을 보여주고, 피보험자는 내용을 기입한다.(E2)(A1)

<사고 정보 입력창>

사고 발생 년-월-일(예시: 2023-01-01) :

사고 발생 시:분(예시: 17:53) :

사고 장소 (예시: 거북골로 3-2 횡단보도) :

사고 유형 (예시: 교통사고) :

사고 개요 (예시: 초록불에 횡단보도를 건너다가 마티즈 차량이 저를 쳤어요) :

3-1. 시스템은 추가적으로 아래와 같이 손괴자 여부 확인 화면을 보여준다.

<손괴자 여부 체크>

해당사고에 손괴자가 존재하나요?

1.네

2.아니요

3-2. 피보험자는 손괴자 여부에 따라 알맞은 버튼을 누른다. 네 버튼을 누를 경우 시스템은 아래의 추가 정보 입력란을 보여준다.

손괴자 이름을 입력하세요 (예시: 홍길동) :

손괴자 전화번호를 입력하세요 (예시: 01012341234) :

4. 시스템은 사고 접수 폼을 랜덤한 사고 접수팀 직원에게 전송한다. -> 테스트 : 현재 로그인 중인 고객으로 일반접수를 여러개 넣어보면 DB에 존재하는 랜덤한 접수팀 직원에게 할당되는 것을 볼 수 있다.

(고객 id: [접수고객의 id]) [접수고객의 이름] 님의 사고가 접수되었습니다.

<사고정보>

사고 일시: [피보험자가 입력한 일시]

사고 장소: [피보험자가 입력한 장소]

사고 유형: [피보험자가 입력한 유형]

사고 개요: [피보험자가 입력한 개요]

손괴자 이름: [피보험자가 입력한 손괴자이름] -> 존재할 경우에만 표시

손괴자 전화번호: [피보험자가 입력한 손괴자 전화번호] -> 존재할 경우에만 표시

1. 확인(접수 승인)

2. 거절

5. 사고 접수팀 직원은 '확인' 버튼을 누른다.(A2)

5-1. 시스템은 피보험자에게 “고객님의 [해당 사고의 일시]에 발생한 사고의 접수가 완료되었습니다.” 메시지를 띄운다.

5-2. 시스템은 해당 접수팀 직원의 사고 조회 메뉴에 해당 사건을 등록하고, 3번에서 피보험자가 접수시 기입한 정보들을 저장한다.

6. 시스템은 사고 처리 현황에서 해당 사건의 상태를 '접수 완료'로 설정한다. (A3)

- A1, 사고 일시가 접수 일시 기준으로 5년이 넘을 경우

1. 시스템은 “발생한지 5년이 넘은 사고는 접수할 수 없습니다” 메시지를 출력하며 접수를 취소시킨다.

- A2, 접수팀 직원이 접수를 거절할 경우

1. 접수팀 직원은 거절 버튼을 누르고 사유를 입력하고 전송 버튼을 누른다.

2. 보험사 시스템은 접수가 거절되었다는 메시지 “고객님의 사고 접수 요청이 거절되었습니다. 거절 사유 : [접수팀 직원이 입력한 사유]”를 피보험자에게 전송한다.

3. 시스템은 해당 사건의 상태를 ‘접수 거절’ 로 설정한다.(A2-A1)

- A2-A1, 접수 거절 판정 상태에서 5년이 지날 경우

1. 사고 접수팀 직원이 초기 메뉴에서 ‘사고 조회’메뉴를 클릭하는 시점에 시스템은 해당 사건을 삭제한다.

- A3, 피보험자가 사고를 취소하려는 경우

1. 피보험자는 사고 조회 메뉴에 들어간다.

2. 시스템은 해당 피보험자가 접수한 사고들을 목록으로 보여준다.

3. 피보험자는 취소할 사건을 선택한다.

3-1. 시스템은 [Basic Path] 4번에 해당하는 정보와 사건의 상태를 보여주고,

접수 취소하기, 나가기 버튼을 보여준다.

3-2. 피보험자는 접수 취소하기 버튼을 누른다.

4. 보험사 시스템은 취소 메시지 "사고번호 [해당 사건 id]의 접수가 취소되었습니다."를 고객에게 보여준다.

5. 보험사 시스템은 취소 메시지 "사고번호 [해당 사건 id]의 접수가 취소되었습니다."를 해당 사고를 접수한 사고접수팀 직원에게 보여준다.

6. 시스템은 사고 처리 현황에서 해당 사건을 삭제한다.

- E1, 화면이 7초이상 표시되지 않을 경우

테스트 : UI/Main.java 의 showCustomerMenu 메소드 첫 줄에 주석처리된 코드를 주석 해제하면 발동.

1. 시스템은 아래의 메시지를 보여주며 로그아웃 시킨다.

[Error!] 현재 시스템의 성능 저하로 서비스가 원활하게 진행되지 못하고 있습니다. 잠시후 다시 시도해 주세요.

고객센터 번호 : 02-8282-7575

- E2, 잘못된 날짜 형식을 입력할 경우

1. 시스템은 "날짜 형식이 올바르지 않습니다. 다시 입력해주세요" 메시지를 띄운다.

2. 피보험자는 날짜를 다시 입력한다.

Usecase Name : 긴급 접수한다.

테스트 : 마찬가지로 1번(고객 로그인) -> ics2023으로 로그인 후 아래 시나리오대로 진행

Description : 고객이 부상 등 긴급한 사유로 인해 자세한 사고 정보를 직접 입력할 수 없어 긴급 접수를 하는 액션이다.

- Basic Path

1. 피보험자는 보험 시스템 어플리케이션 초기화면(사고 접수, 사고 조회 메뉴 존재)에서 사고 접수 메뉴에 들어간다. (E1)

1. 피보험자는 긴급 접수하기 버튼을 클릭한다. (A1)

2. 시스템은 사고 접수팀,현장 조사팀에게 메시지를 전송한다.

3.사고 접수팀 직원과 현장 조사팀 직원은 아래와 같은 긴급 접수가 들어왔다는 메시지를 받는다.

(고객 id: [접수고객의 id]) [접수고객의 이름]님의 사고가 긴급 접수되었습니다.

<사고 정보>

사고 일시: [접수한 날짜 및 시간 자동 입력]

사고 장소: [접수고객의 GPS 장소 자동 입력]

접수자 이름: [접수고객의 이름 자동 입력]

접수자 전화번호: [접수고객의 전화번호 자동 입력]

3-1.시스템은 출동중이 아닌 현장 조사팀 직원에게 해당 긴급 접수를 자동으로 할당한다. (A2)

4.시스템은 '사고 조회' 메뉴에 해당 사건의 상태를 '접수 완료(출동 대기)' 등록하고, 위의 정보들을 저장한다. (A3)

테스트 : 고객으로 접속중이라면 로그아웃 후 2번(직원 로그인) -> 해당 사고를 접수받은 현장조사팀 직원 id(ie2023 혹은 ie2024)로 로그인 후 아래 시나리오대로 진행

5.현장 조사팀 직원은 '사고 조회'메뉴에서 자신에게 할당된 사건을 클릭하고 출동하기 버튼을 누른다.

5-1.시스템은 해당 사건의 상태를 '접수 완료(출동 중)'으로 바꾼다

6. 시스템은 해당 사고를 접수한 피보험자에게 “현장 조사팀이 출동했습니다”메세지를 보낸다. (A4)

7. 현장 조사팀 직원은 현장에서 직접 정보를 조사하여 '사고 조회'메뉴에서 해당 사고를 클릭한다.

8. 현장 조사팀 직원은 [Basic Path]의 3번~의 정보를 대신 기입하여 사건을 등록한다.(시간 정보는 긴급접수한 시점으로 자동완성) (A5)

9. 시스템은 사고 처리 현황에서 해당 사건의 상태를 '접수 완료'로 설정한다.

- A1, 이미 접수 처리중인 긴급접수가 있는 고객이 또다시 긴급접수할 경우

시스템은 고객님의 접수한 긴급 사건번호 [고객이 이전에 접수한 사건 id]에 대한 긴급 접수가 아직 처리 중입니다. 이전의 사건이 접수 완료되면 긴급 접수해주세요. 메세지를 출력하고, 접수를 막는다.

- A2, 모든 현장 조사팀 직원이 출동중일 경우

1. 시스템은 “현재 모든 현장 조사팀 직원이 출동중입니다. 조속히 처리하겠습니다” 메세지를 피보험자에게 전송한다. 이후 A5에서 그 처리가 이루어진다.

- A3, 피보험자가 현장 조사팀의 출동 전에 긴급 접수를 취소하려는 경우

1. 피보험자는 사고 조회 메뉴에 들어간다.

2. 시스템은 해당 피보험자가 접수한 사고들을 목록으로 보여준다.

3. 피보험자는 취소할 사건을 선택한다.

3-1. 시스템은 [Basic Path] 3번에 해당하는 정보와 사건의 상태를 보여주고,
접수 취소하기, 나가기 버튼을 보여준다.

3-2. 피보험자는 접수 취소하기 버튼을 누른다.

4. 보험사 시스템은 취소 메시지 "사고번호 [해당 사건 id]의 긴급 접수가 취소되었습니다."를
고객에게 보여준다.

5. 보험사 시스템은 취소 메시지 "사고번호 [해당 사건 id]의 긴급 접수가 취소되었습니다."를
사고접수팀 직원, 현장조사팀 직원에게 보여준다.

6. 시스템은 사고 처리 현황에서 해당 사건을 삭제한다.

- A4, 출동이 시작되고 피보험자가 긴급 접수를 취소할 경우

테스트 : 2번(직원 로그인) -> ce2023으로 로그인 후 아래 시나리오대로 진행

0. [A3] 1~3-2 까지 동일하게 진행된다.

1. 시스템은 "이미 현장 조사 직원들이 출발하여 취소시 수수료가 발생합니다." 라는 메시지를
피보험자에게 전송한다.

2. 피보험자는 해당 메시지를 확인하고 '확인' 버튼을 누른다.

3. 보험사 시스템은 취소 메시지 "사고번호 [해당 사건 id]의 긴급 접수가 취소되었습니다."를
고객에게 보여준다.

4. 보험사 시스템은 취소 메시지 "사고번호 [해당 사건 id]의 긴급 접수가 취소되었습니다."를
사고접수팀 직원, 현장조사팀 직원에게 보여준다.

5. 시스템은 사고 목록에서 해당 사건을 삭제한다.

6. 시스템은 피보험자의 수수료에 30000원을 추가시킨다.

- A5, 모든 현장 조사 직원이 사건을 할당받아서 현장 조사팀이 배치되지 않은 사건이 있을 경우

1. 시스템은 현장 접수 직원이 사건을 등록해서 해당 사건에 대한 처리를 마치고 담당
사건이 없어지는 시점에 자동적으로 밀린 사건을 해당 직원에게 할당시킨다. -> 테스트 :
(고객 로그인) ics2023으로 로그인해 긴급 접수시키고, (고객 로그인) ics2024로 다시 로그인해
긴급 접수시키면 현재 더미데이터에서 현장조사팀은 2명이므로 모든 요원이 출동중인 상황이
된다. 이 상황에서 (고객 로그인) ics2025로 로그인해 긴급접수를 하게되면 이 긴급접수는
대기상태에 있게 되는데, (직원 로그인) ie2023이나 ie2024 둘 중 한명으로 로그인해서

사고조회-접수를 눌러서 정보를 기입하여 기존의 자신이 담당하던 긴급접수를 해결하는 시점에 밀려있던 ics2025의 긴급접수가 즉시 자신에게 할당되는 모습을 볼 수 있다.

2. [Basic Path] 5번부터 동일하게 진행한다.

- E1, 화면이 7초이상 표시되지 않을 경우

1. 시스템은 아래의 메시지를 보여주며 로그아웃 시킨다.

[Error!] 현재 시스템의 성능 저하로 서비스가 원활하게 진행되지 못하고 있습니다. 잠시후 다시 시도해 주세요.

고객센터 번호 : 02-8282-7575

Usecase Name : 보상금 지급여부를 결정한다

Description : 보상팀 직원이 피보험자와 피보험자가 든 보험 정보, 사고 정보 등을 검토하고, 검토한 결과를 토대로 보상금을 지급할지에 대한 여부를 결정하는 액션이다. '사고를 접수한다'유스케이스와 include 관계로 설정된 이유는 보상금 지급여부를 결정하려면 반드시 사고 접수가 선행되어야 하기 때문이다.

테스트 : 2번(직원 로그인) -> ce2023으로 로그인 후 아래 시나리오대로 진행

- Basic Path

0. 시스템은 보상팀 직원에게 아래와 같은 메뉴들을 보여준다. (E1)

서류 검토하기

보상 여부 결정하기

보상금 책정하기

보상금 지급하기

구상 신청하기

1. 보상팀 직원은 '보상금 지급하기' 메뉴를 클릭한다.

2. 시스템은 현재 접수된 사건 목록들을 보여준다. 이 때, 보상 담당자가 미배정된 사건들과 해당 보상팀 직원이 배정하고 있는 사건들을 아래와 같이 보여준다. (A0)

1. [미배정] 사고번호: [미배정된 사건의 id]
2. [거절됨] 사고번호: [해당 보상팀 직원이 지급여부를 거절시킨 사건의 id]
3. [추가서류 요청됨] 사고번호: [해당 보상팀 직원이 추가서류를 요청한 사건의 id]

3. 보상팀 직원은 목록에서 해당 사고를 클릭한다.

4. 시스템은 아래의 정보를 보여준다(손괴자 존재하는 사건일 경우 손괴자 이름과 전화번호도 목록에 표시한다.)

사고 번호: [해당 사건의 id]

사고 일시: [해당 사건의 일시]

사고 장소: [해당 사건의 장소]

사고 유형: [해당 사건의 유형]

사고 개요: [해당 사건의 개요]

피보험자 이름: [해당 사건 접수한 피보험자의 이름]

피보험자 전화번호: [해당 사건 접수한 피보험자의 전화번호]

처리 상태: [해당 사건의 상태]

4-1.시스템은 '나가기'버튼과 '고객 정보 검토하기'버튼을 보여주고, 보상팀 직원은 고객 정보 검토하기 버튼을 누른다.

4-2.시스템은 아래와 같이 고객 정보와 , 고객의 계약 정보들을 나열한다.

<고객 정보>

id: [접수고객의 id]

이름: [접수고객의 이름]

주민번호: [접수고객의 주민번호]

나이: [접수고객의 나이]

성별: [접수고객의 성별]

전화번호: [접수고객의 전화번호]

직업: [접수고객의 직업]

가족력: [접수고객의 가족력]

건강검진서: [접수고객의 건강검진서]

가족관계서 : [접수고객의 가족관계서]

재직증명서 : [접수고객의 재직증명서]

<계약 정보>

계약 id: [접수고객이 계약한 계약의 id]

보험 id: [접수고객이 계약한 보험의 id]

보험 상품명: [접수고객이 계약한 보험의 상품명]

보험 설명: [접수고객이 계약한 보험의 설명]

보상금 지급 여부 승인하기

보상금 지급 여부 거절하기

추가 서류 요청하기

5. 보상팀 직원은 '보상금 지급 여부 승인하기' 버튼을 클릭한다.

(A1)(A2)

5-1. 시스템은 사고 처리 현황에서 해당 사건의 상태를 '지급여부 승인'으로 변경한다.

6. 시스템은 피보험자에게 "고객님이 접수하신 사고번호 : [[해당 사고 id]] 대한 보상금 지급여부가 승인되었습니다. 이제 보상금 책정 절차에 들어갑니다." 라는 메시지를 전송한다.

7. 해당 사건에 현재 보상팀 직원이 배정된다.

- A0, 조회할 사건이 없을 시

시스템은 "처리중인 사건이 없습니다"라는 화면을 보여주고 보상팀 직원을 메인 메뉴로 이동시킨다.

- A1, 보상금 지급여부 거절하기 버튼 클릭 시

1. 시스템은 거절 사유를 입력하세요: 라는 입력란을 보여준다.

2. 보상팀 직원은 거절 사유를 입력한다.

2-1. 시스템은 피보험자에게

"고객님이 접수하신 사고번호 : [[해당 사고 id]] 대한 보상금 지급여부가 거절되었습니다.

거절 사유 : [보상팀 직원이 입력한 거절 사유]

추가적인 문의나 이의가 있으시면 담당자에게 문의해주시기 바랍니다.

담당자 이름: [보상팀 직원의 이름], 담당자 전화번호 : [보상팀 직원의 전화번호]"

메시지를 전송한다

3. 시스템은 해당 사건의 상태를 '지급여부 거절'으로 바꾼다.(A1-A1)(A2-A2)

4. 해당 사건의 보상팀 직원이 배정된다(데이터베이스에 해당 직원id 저장)

- A1-A1, 지급여부 거절 상태에서 접수일 기준 5년이 지날 경우

1. 시스템은 보상팀 직원이 사고를 조회하려고 [Basic Path]의 0번을 수행할 때 해당 사건을 삭제한다.

- A1-A2, 피보험자가 전화 통화로 이의를 제기하고, 이의가 승낙되었을 경우

1. 보상팀 직원은 '보상 여부 결정하기'메뉴에서 해당 사건을 클릭한다.

2.시스템은 [Basic Path] 4번과 같은 정보를 보여준다.

3.보상팀 직원은 보상금 지급 여부 승인하기 버튼을 누른다.

4.시스템은 해당 사고의 상태를 '지급여부 승인'으로 변경한다.

- A2, 추가 서류 요청하기 버튼 클릭 시

1.시스템은 “추가적으로 요청할 서류들을 입력하세요”라는 문구와 아래의 입력란을 보여준다

서류 이름: [입력란]

다른 서류도 요청하기

요청 완료하기

2.보상팀 직원은 필요한 서류만큼 다른 서류도 요청하기 버튼을 눌러 입력한다.

3.요청할 서류를 모두 기입하고 요청 완료하기 버튼을 누른다.

4.시스템은 “고객님이 접수하신 사고 번호 [[해당 사고의 id]] 서류가 요청되었습니다. 추가 서류: [보상팀 직원이 요청한 서류들]

해당 서류(들)는 [보상팀 직원의 이메일 주소]로 보내주시길 바랍니다.”

메세지를 전송한다.

5.시스템은 해당 사건(사고)의 상태를 '추가서류 요청[[요청된 서류들]]'로 변경한다.

6. 해당 사건의 보상팀 직원이 배정된다(데이터베이스에 해당 직원id 저장)

7.고객이 이메일을 통해 제출을 확인했다면 보상팀 직원은 직접 [Basic Path] 의 0번부터의 과정을 다시 수행한다.

- E1, 화면이 7초 이상 표시되지 않을 경우

테스트 : UI/Main.java 의 showEmployeeMenu 메소드 첫 줄에 주석처리된 코드를 주석 해제하면 발동.

1. 시스템은 아래의 메세지를 보여주며 로그아웃 시킨다.

[Error!] 현재 시스템의 성능 저하로 서비스가 원활하게 진행되지 못하고 있습니다. 잠시후 다시 시도해 주세요.

전산팀 사내 유선번호 :047

Usecase Name : 보상금을 책정한다

Description : 보상팀 직원이 피보험자에게 최종적으로 얼마만큼의 금액을 보상금으로 지급할 것인지 산정하는 액션이다. 이 과정에서 피보험자는 진료비 계산서, 손해 계산서를 전송하여 보상금 책정에 필요한 정보를 제공하며, 보상팀 직원은 이를 토대로 보상금 액수를 결정한다. 보상금을 지급여부를 결정한다 유스케이스가 먼저 선행되어야 하기에 include 관계로 이어져있다.

테스트: 2번(직원 로그인) -> ce2023으로 로그인 후 아래 시나리오대로 진행

- Basic Path

0. 시스템은 보상팀 직원에게 아래와 같은 메뉴들을 보여준다. (E1)

서류 검토하기

보상 여부 결정하기

보상금 책정하기

보상금 지급하기

구상 신청하기

1. 보상팀 직원은 '보상금 책정하기' 메뉴를 클릭한다.

2. 시스템은 현재 보상팀 직원이 보상금 책정이 가능한(이전 유스케이스가 완료된 상태) 동시에 현재 조회하려는 직원이 담당하고 있는 사건들만 필터링하여 목록으로 보여준다. 이 때, 목록의 앞부분에 아래와 같이 현재 사건의 상태가 간략하게 표시된다.(A0)

[미책정] 1. 사고번호 : [사고id]

[서류 요청중] 2. 사고번호 : [사고id] ->현재 시나리오 8-1번을 수행하고 난 이후

[서류 도착] 3. 사고번호 : [사고id] ->현재 시나리오 14번을 수행하고 난 이후

3. 보상팀 직원은 처리할 사건을 선택한다.

4. 시스템은 아래의 정보를 보여준다(손괴자 존재하는 사건일 경우 손괴자 이름과 전화번호도 목록에 표시한다.)

사고 번호: [해당 사건의 id]

사고 일시: [해당 사건의 일시]

사고 장소: [해당 사건의 장소]

사고 유형: [해당 사건의 유형]

사고 개요: [해당 사건의 개요]

피보험자 이름: [해당 사건 접수한 피보험자의 이름]

피보험자 전화번호: [해당 사건 접수한 피보험자의 전화번호]

처리 상태: [해당 사건의 상태]

4-1.시스템은 '나가기'버튼과 '고객 정보 검토하기'버튼을 보여주고, 보상팀 직원은 고객 정보 검토하기 버튼을 누른다.

5. 시스템은 아래와 같이 고객의 정보와 고객의 계약 정보들을 보여준다.

<고객 정보>

id: [해당 사고를 접수한 고객 id]

이름: [해당 사고를 접수한 고객 이름]

주민번호: [해당 사고를 접수한 고객 주민번호]

나이: [해당 사고를 접수한 고객 나이]

성별: [해당 사고를 접수한 고객 성별]

전화번호: [해당 사고를 접수한 고객 전화번호]

직업: [해당 사고를 접수한 고객 직업]

가족력:

[해당 사고를 접수한 고객 가족력 정보]

-건강검진서-

[해당 사고를 접수한 고객 건강검진서]

-가족관계서-

[해당 사고를 접수한 고객 가족관계서]

-재직증명서-

[해당 사고를 접수한 고객 재직증명서]

-진료비 계산서-

[해당 사고를 접수한 고객 진료비 계산서] -> 없을 시 '없음'

-손해 계산서-

[해당 사고를 접수한 고객 건강검진서] -> 없을 시 '없음'

<계약 정보>

계약 id: [해당 고객이 계약한 계약의 id]

보험료: [해당 고객이 계약한 보험료]

보험 id: [해당 고객이 계약한 보험의 id]

보험 상품명: [해당 고객이 계약한 보험의 상품명]

보험 설명: [해당 고객이 계약한 보험의 설명]

6. 시스템은 보상금 책정하기, 보상금 책정 관련 서류 요청하기, 나가기 버튼을 보여준다.

7. 보상팀 직원은 보상금 책정 관련 서류 요청하기 버튼을 클릭한다 (A1)

8. 시스템은 해당 사고의 피보험자에게 “고객님이 접수하신 사고 번호 [[해당 사고 id]]의 보상금이 책정되기 위한 서류 요청이 도착했습니다. 사고 조회 메뉴에서 해당 사고를 클릭하고 서류 업로드하기 버튼을 클릭해주세요.”메세지를 전송한다.

8-1. 시스템은 해당 사건의 상태를 ‘책정서류 요청중’으로 변경한다. (A2)

테스트 : 직원으로 로그인 중이라면 로그아웃 후 1번(고객 로그인) -> ics2023(해당 사고 접수한 고객 id)으로 로그인 후 아래 시나리오대로 진행

9. 피보험자는 시스템에 로그인하여 사고 접수, 사고 조회 메뉴 중 사고 조회를 클릭한다.

10. 시스템은 해당 피보험자가 접수한 사고 번호 목록들을 보여주고, 피보험자는 서류를 제출해야 하는 사건을 클릭한다.

10-1. 시스템은 아래와 같이 사고의 세부정보를 보여준다

사고 일시: [해당 사고 일시]

사고 장소: [해당 사고 장소]

사고 유형: [해당 사고 유형]

사고 개요: [해당 사고 개요]

처리 상태: 책정서류 요청중

10-2. 시스템은 나가기 버튼과 책정 관련 서류 제출하기 버튼을 보여준다

11. 피보험자는 책정 관련 서류 제출하기 버튼을 클릭한다.

12. 시스템은 아래와 같은 업로드 창을 보여준다.

<보상금 책정 관련 서류 업로드>

두개의 서류 중 필요한 서류를 업로드해주세요 (예시: 자동차 사고 - 다친 부분에 대한 진료비
계산서, 자동차에 대한 수리 견적이 담긴 피해 계산서 모두 첨부)

진료비 계산서가 존재합니까?

1.예

2.아니요

13. 피보험자는 '예'를 누른다. ('아니요' 누를시 해당 서류의 업로드 패스)

14. 시스템은 진료비 계산서 업로드란(여기에 파일을 업로드)을 보여주고 피보험자는 이곳에
진료비 계산서를 업로드한다. (E2) -> 테스트: 사전에 준비된 서류 파일인 'medicalBill'을 입력

15. 시스템은 아래의 화면을 보여준다.

피해 계산서가 존재합니까?

1.예

2.아니요

15. 피보험자는 마찬가지로 피해 계산서 파일을 업로드한다. (E2)(A3)

-> 테스트: 사전에 준비된 서류 파일인 'damageBill'을 입력

16. 시스템은 피보험자에게 "보상금 책정 관련 서류 요청이 완료되었습니다"메세지를 보여주고,
해당 사건의 담당 보상팀 직원에게 "사고 번호 [[사고 id]] 에 대한 보상금 책정 관련 서류가
제출되었습니다."메세지를 전송한다.

테스트: 2번(직원 로그인) -> ce2023으로 로그인 후 아래 시나리오대로 진행

17. 보상팀 직원은 이 시나리오의 0번~6번을 동일하게 진행하고, 보상금 책정하기 버튼을
클릭한다.

18. 시스템은 보상금을 입력하세요 라는 입력란을 보여주고, 피보험자는 보상금을 입력한다. (E3)

19. 시스템은 아래의 메세지를 해당 사고를 접수한 피보험자에게 전송한다.

고객님이 접수하신 사고 번호 [[해당 사고 id]] 의 보상금이 책정되었습니다.

보상 금액: [보상팀 직원이 책정한 보상금액]

추가적인 문의나 이의가 있으시면 담당자에게 문의해주시기 바랍니다.

담당자 이름: [책정한 보상팀 직원의 이름]

담당자 전화번호: [책정한 보상팀 직원의 전화번호]

20. 시스템은 해당 사고의 보상금을 측정한 양만큼 설정시키고, 해당 사고의 상태를 ‘보상금 책정완료’로 설정한다.

- A0, 조회할 사건이 없을 시

시스템은 “처리중인 사건이 없습니다”라는 화면을 보여주고 보상팀 직원을 메인 메뉴로 이동시킨다.

- A1, 해당 사고를 접수한 피보험자의 진료비 계산서, 손해 계산서가 모두 없는데 ‘보상금 책정하기’ 버튼을 누를 경우

1. 시스템은 아래와 같은 경고 메시지를 띄운다.

[Warning!] 보상금 책정 관련 서류 정보가 없습니다. 먼저 서류를 요청하고 이를 검토한 후 보상금을 책정해주세요

2. 시스템은 다시 보상금 책정하기, 보상금 책정 관련 서류 요청하기, 나가기 버튼을 보여준다.

- A2, 만일 피보험자가 사고 일시로부터 5년이 넘도록 서류를 제출하지 않았을 경우.

1. 시스템은 피보험자에게 "고객님이 접수하신 사고 번호 [[해당 사고번호 id]]의 보상금 책정 관련 서류를 5년동안 제출하지 않아 청구권 만료로 해당 사건의 보상처리가 종료되었습니다."메세지를 전송한다.

2. 시스템은 해당 사건을 삭제한다. -> 위 과정은 [Basic Path]1번을 수행할때 자동 업데이트된다.

- A2, 피보험자가 아무런 서류도 업로드하지 않았을 경우

1. 시스템은 “어떠한 파일도 업로드하지 않아 메인메뉴로 이동합니다.”

메세지를 보여주며 메인메뉴로 이동한다.

- E1, 화면이 7초 이상 표시되지 않을 경우

1. 시스템은 아래의 메시지를 보여주며 로그아웃 시킨다.

[Error!] 현재 시스템의 성능 저하로 서비스가 원활하게 진행되지 못하고 있습니다. 잠시후 다시 시도해 주세요.

전산팀 사내 유선번호 :047

- E2, 잘못된 확장자의 파일을 업로드할 경우

1. 시스템은 ‘[Error!] 잘못된 파일입니다. 다시 업로드해주세요’라는 메시지를 보여주고 피보험자에게 다시 업로드하도록 한다.

- E3, 잘못된 금액 형식을 입력할 경우(문자 등)

1.시스템은 '[Error!] 올바른 금액을 입력하세요'라는 메시지를 보여주고 보상팀 직원이 다시 금액을 올바르게 입력하도록 한다.

Usecase Name : 보상금을 지급한다.

Description : 보상팀 직원이 책정한 보상금을 고객의 계좌로 입금시키는 액션이다. '보상금을 책정한다'유스케이스와 include 관계로 형성된 이유는 보상금을 지급하려면 보상금을 책정해야 하기 때문이다.

테스트: 2번(직원 로그인) -> ce2023으로 로그인 후 아래 시나리오대로 진행

- Basic Path

0. 시스템은 보상팀 직원에게 아래와 같은 메뉴들을 보여준다. (E1)

서류 검토하기

보상 여부 결정하기

보상금 책정하기

보상금 지급하기

구상 신청하기

1. 보상팀 직원은 '보상금 지급하기' 메뉴를 클릭한다.

2. 시스템은 현재 접수된 사건 목록들중 상태가 '보상금 책정완료'이면서 현재 로그인한 보상팀 직원이 담당하고 있는 사건들만 필터링하여 보여준다.

1. [미지급] 사고번호: [보상금이 미지급된 사건의 id]

2. [지급 보류] 사고번호: [해당 보상팀 직원이 지급을 보류시킨 사건의 id]

3. 보상팀 직원은 처리할 사건을 선택한다.

4. 시스템은 아래의 정보를 보여준다(손괴자 존재하는 사건일 경우 손괴자 이름과 전화번호도 목록에 표시한다.)

사고 번호: [해당 사건의 id]

사고 일시: [해당 사건의 일시]

사고 장소: [해당 사건의 장소]

사고 유형: [해당 사건의 유형]

사고 개요: [해당 사건의 개요]

피보험자 이름: [해당 사건 접수한 피보험자의 이름]

피보험자 전화번호: [해당 사건 접수한 피보험자의 전화번호]

보상 책정금: [해당 사건의 보상 책정금]

처리 상태: [해당 사건의 상태]

4-1. 시스템은 나가기, 보상금 지급하기, 보상금 지급 보류하기, 보상금 재책정하기 버튼을 보여주며, 보상팀 직원은 보상금 지급하기 버튼을 클릭한다.(A1)(A2)

5. 시스템은 아래와 같은 송금창을 띄워준다.

<보상금 지급>

받을 사람: [해당 사건을 접수한 고객 이름]

받을 계좌번호: [해당 사건을 접수한 고객 계좌번호]

보내는 사람: (주)iGO

보내는 계좌번호: 94320200299734

비고: iGO 보상처리팀 [해당 사건 담당 보상팀 이름][직급]

5-1. 시스템은 아래와 같은 문구와 버튼을 보여준다.

송금하시겠습니까?

1.예

x.취소하기 -> 메인 메뉴로 나감

5-2. 보상팀 직원은 '예'버튼을 누른다(E2)

6. 시스템은 해당 사건의 상태를 '보상금 지급완료'로 변경하고 '입금이 완료되었습니다.'문구를 보여준다.

7. 시스템은 해당 사고를 접수한 피보험자에게 아래와 같은 메시지를 전송한다.

고객님의 사건번호 [[해당 사고 id]] 에 대한 보상금 [송금된 보상금]가 고객님의 계좌 [피보험자의 계좌번호]로 입금되었습니다.

8. 메인 메뉴로 이동한다.

- A1, 보상금 지급 보류하기 버튼을 클릭할 경우

1.시스템은 아래와 같은 보류 사유와 보류 예상일 입력란을 보여준다.

<보상금 지급 보류>

보류 사유를 입력하세요:

보류 예상일을 입력하세요:

2. 시스템은 아래와 같은 메시지를 해당 피보험자에게 전송한다.

고객님의 사건번호 [[해당 사고 id]] 에 대한 보상금 지급이 보류되었습니다.보류 사유: [보상팀 직원이 입력한 보류사유], 보류 예상일: [보상팀 직원이 입력한 예상 지체일]

2-1.시스템은 해당 사건의 상태를 '보상금 지급보류'로 변경한다.

A2, 보상금 재책정하기 버튼을 눌렀을 경우

1. 시스템은 아래와 같은 메시지를 해당 피보험자에게 전송한다.

고객님의 사건번호 [[해당 사고 id]] 에 대한 보상금이 다시 책정될 예정입니다.

2. 시스템은 해당 사건의 상태를 '지급여부 승인'으로 바뀌서 보상금을 책정한다 유스케이스로 돌아가게 한다.

- E1, 화면이 7초동안 표시되지 않을 경우

[Error!] 현재 시스템의 성능 저하로 서비스가 원활하게 진행되지 못하고 있습니다. 잠시후 다시 시도해 주세요.

전산팀 사내 유선번호 :047

- E2, 현재 시각이 은행 점검시간일 경우

1. 시스템은 보상팀 직원에게 아래와 같은 문구를 보여준다.

[Error!]: 은행 점검시간입니다.(00:00~01:30) 점검이 끝나면 다시 시도해주세요

Usecase Name : 손해배상 청구(구상)를 한다

Description : 구상이란 보상팀 직원이 피보험자에게 보상금을 지급한 이후, 만일 피보험자가 누군가에게 피해를 입었다면 피해를 입힌 당사자에게 보상금의 일부 혹은 그 이상의 금액을 청구하여 보상금 지출을 메꾸는 것을 말한다. '보상금을 지급한다' 와 include 관계로 설정된 것은 구상 청구를 하려면 우선 피보험자에게 보상금을 지급해야 하기 때문이다.

테스트 : 2번(직원 로그인) -> ce2023으로 로그인 후 아래 시나리오대로 진행

- Pre-Condition : 보상금 지급이 완료되고, 해당 사건에 손괴자가 존재하는 사건
- Basic Path

0. 시스템은 보상팀 직원에게 아래와 같은 메뉴들을 보여준다. (E1)

서류 검토하기

보상 여부 결정하기

보상금 책정하기

보상금 지급하기

구상 신청하기

1. 보상팀 직원은 '보상금 지급하기' 메뉴를 클릭한다.
2. 시스템은 현재 접수된 사건 목록들중 상태가 보상금 지급이 완료된 이후의 사건이면서 현재 로그인한 보상팀 직원이 담당하고 있는 사건들만 필터링하여 보여준다. 아래처럼 해당 사건의 상태에 따라 앞에 태그에 붙는 문구가 달라진다.(AO)

1. [미구상] 사고번호: [구상 절차가 아직 시작되지 않은 사건의 id]
2. [구상 처리중] 사고번호: [구상 절차가 진행되고있는 사건의 id]
3. [구상 완료] 사고번호: [구상 절차가 완료된 사건의 id]

3. 보상팀 직원은 처리할 사건을 선택한다.

4. 시스템은 아래의 정보를 보여준다

사고 번호: [해당 사건의 id]

사고 일시: [해당 사건의 일시]

사고 장소: [해당 사건의 장소]

사고 유형: [해당 사건의 유형]

사고 개요: [해당 사건의 개요]

피보험자 이름: [해당 사건 접수한 피보험자의 이름]

피보험자 전화번호: [해당 사건 접수한 피보험자의 전화번호]

피보험자 진료비 계산서 : ['보상금을 책정한다'유스케이스에서 피보험자가 업로드했던 진료비
계산서 내용] -> 제출 안했다면 없음으로 표시

피보험자 피해 계산서 : ['보상금을 책정한다'유스케이스에서 피보험자가 업로드했던 피해 계산서
내용] -> 제출 안했다면 없음으로 표시

손괴자 이름: [해당 사건의 손괴자 이름]

손괴자 전화번호 : [해당 사건의 손괴자 전화번호]

보상 책정금: [해당 사건의 보상 책정금]

처리 상태: [해당 사건의 상태]

- 4-1. 시스템은 아래와 같은 버튼을 보여준다.

구상금 청구서 발송 -> [미구상]인 사건들에만 보여짐

구상금 납부기한 연장 -> [구상 처리중]인 사건들에만 보여짐

구상 완료 처리 -> [구상 처리중]인 사건들에만 보여짐

x. 뒤로가기

5. 보상팀 직원은 위의 정보를 확인하여 손괴자에게 전화하여 구상금 청구가 될 것임을 고지한다.

5-1. 보상팀 직원은 고지한 이후 '구상금 청구서 발송'버튼을 누른다.

6. 시스템은 아래와 같은 구상금 청구서 작성화면을 보여준다.

사고 일시: [해당 사고 일시로 자동입력]

사고 장소: [해당 사고 장소로 자동입력]

사고 개요: [해당 사고 개요로 자동입력]

피해자 이름: [해당 사고 접수한 피보험자 이름으로 자동입력]

피해자 전화번호: [해당 사고 접수한 피보험자 이름으로 자동입력]

청구금액 :

명세:

납부 계좌: [보험사 납부 계좌번호로 자동입력]

납부 기한 : -> (E2),(E3)

6-1. 시스템은 아래의 버튼을 보여준다

1.SMS 발송

x.뒤로가기

7. 보상팀 직원은 6번에서 비어있는 정보들을 작성하고 'SMS발송'버튼을 누른다.

8. 시스템은 손괴자에게 6번에서 작성된 정보를 SMS를 통해 전송한다. (이 때 손괴자는 우리 시스템의 이용자가 아니기에 UI에 따로 뜨진 않고 SMS가 발송되었습니다 라는 문구만 보상팀 직원에게 표시된다.)

9. 손괴자는 청구서에 따라 금액을 납부한다. (A1)(A2)

10. (손괴자는 보험사 시스템의 사용자가 아니기에 시스템은 손괴자가 입금시 보상팀 직원에게 알림을 보낼 수 없다. 따라서 입금 여부를 직접 확인해야 한다.) 보상팀 직원은 [Basic Path] 2번에 해당하는 화면에서 해당 사건을 선택하고 '구상 완료 처리'버튼을 누른다.

10-1. 시스템은 보상팀 직원에게 "해당 사건의 구상금 [해당 사건의 구상신청금] 을 입금받았습니까?"메세지와 예 아니오 버튼을 보여준다.

10-2. 보상팀 직원은 '예'버튼을 누른다. -> 아니오 누르면 메인메뉴로

11. 시스템은 "구상 절차가 완료되었습니다"라는 문구를 띄운다.

11-1. 시스템은 해당 사건의 상태를 '구상 완료'로 바꾼다.

- A0, 조회할 사건이 없을 시(이전 유스케이스가 완료되어 보상금 지급을 마치고 손피자가 존재하는 사건이 없을 시)

시스템은 “처리중인 사건이 없습니다”라는 화면을 보여주고 보상팀 직원을 메인 메뉴로 이동시킨다.

- A1, 손피자의 사정으로 납부 기한 연장을 요청할 경우

0. 손피자는 받은 문자메세지 속 기재된 담당자 전화번호를 통해 담당자와 통화하여 납부 기한 연장을 요청한다.

1. 보상팀 직원은 손피자의 요청을 검토한 후, 만일 합당하다고 판단되면 Basic Path 2번에 해당하는 메뉴에 들어가 해당 사건을 클릭한다.

2. 보상팀 직원은 '구상금 납부기한 연장' 버튼을 눌러 납부기한을 수정한다.(E2)

3. 시스템은 납부기한이 수정되었다는 SMS를 손피자에게 전송한다.

4. [Basic Path]의 9번부터 동일하게 진행된다.

- E1, 화면이 7초동안 표시되지 않을 경우

[Error!] 현재 시스템의 성능 저하로 서비스가 원활하게 진행되지 못하고 있습니다. 잠시후 다시 시도해 주세요.

전산팀 사내 유선번호 :047

- E2, 잘못된 날짜형식을 입력할 경우

1.시스템은 “날짜 형식이 올바르지 않습니다. 다시 입력해주세요”라는 메세지를 출력하고 날짜를 다시 입력하게 한다.

- E3, 입력한 납부기한이 사고발생일로부터 5년을 넘어가는 경우.

1.시스템은 "[Error!]구상채권은 5년이 넘을 경우 시효가 소멸됩니다. 5년 이내에서 연장해주세요."라는 메세지를 출력하여 해당 기간을 입력하지 못하도록 막고 다시 입력하게 한다.

Usecase Name : 보상금 내역을 기록한다.

(비고 : 미구현 결정)

이유: 시스템 밖에서 주로 업무를 처리하는 부서(회계팀) 위주의 유스케이스 - 한정된 개발 시간에서 중요도 낮음)

Description : 보상금 지급 및 구상처리가 완료된 후 회계팀이 보상금으로 얼마를 지출하고 구상금으로 얼마를 받았는지 장부에 기록하고, 보상팀 직원은 보상 처리 절차를 마무리하는

액션이다. '보상금을 지급한다'와 include 관계로 형성된 이유는 보상금이 지급이 완료되어야만 이를 기록할 수 있기 때문이다. '손해배상 청구(구상)를 한다'와 extend 관계인 이유는 보상금 내역을 기록하는데 있어서 해당 사건의 손괴자 여부에 따라 구상을 할 수도, 안 할 수도 있기 때문이다.

- **Pre-condition** : 손괴자가 없는 사건의 경우 보상금 지급이 완료된 상태, 손괴자가 있는 사건의 경우 구상 청구가 완료된 상태(소송이 있었을 경우 소송까지 완료된 상태)

- **Basic Path**

모든 시나리오에 적용 : E1

0. 위에 기재된 유스케이스의 초반부와 같이 보상팀은 사건번호를 입력하여 해당 사건을 조회한다.('보상금 지급여부를 결정한다' 유스케이스의 Basic Path 1~4번과 동일)

1. 보상팀 직원은 '보상금 내역 기록' 버튼을 클릭한다.(E2)

2. 시스템은 보상금 액수, 구상 요청했던 사건시 구상금 액수를 자동완성하여(각 절차를 밟으며 사건정보에 저장해놔던 금액을 불러온다)보상팀 직원에게 최종적으로 확인하도록 해당 액수들을 보여준다.(A1)

3. 보상팀 직원은 '확인' 버튼을 클릭한다.

4. 시스템은 해당 금액 정보들을 회계팀 직원에게 전송한다.

5. 회계팀 직원은 해당 정보를 토대로 회계장부를 업데이트한다.

6. 보상팀 직원은 '사고 조회' 메뉴에 들어가 해당 사건의 상태를 '종결'로 바꾼다.(종결은 이전 유스케이스들처럼 시나리오가 종료되는 시점에 시스템이 상태를 자동으로 변경해주지 않는다. 마무리짓는 과정이기에 직원이 직접 이를 종결시켜야 한다.)

- **A1, 구상 소송을 진행했던 사건인 경우**

1. 시스템은 자동으로 입력란에 소송 패소 여부와 금액(소송 비용 & 승소시 승소 획득금, 패소시 패소 지출금)을 적는 칸을 추가적으로 보여주고 해당 금액들을 채워넣는다.

[Basic Path]의 3번부터 동일하게 진행된다.

- **E1, 화면이 7초 이상 표시되지 않을 경우**

[Error!] 현재 시스템의 성능 저하로 서비스가 원활하게 진행되지 못하고 있습니다. 잠시후 다시 시도해 주세요.

전산팀 사내 유선번호 :047

- E2, 해당 사건의 상태가 '보상금 지급 완료'나 '구상 완료', '구상 완료(소송 거절)'가 아닌 경우

1. 시스템은 "해당 사건은 아직 처리가 완료되지 않은 상태입니다! 모든 처리를 완료하고 다시 시도해주세요" 메시지를 출력한다.

Usecase Name : 구상 소송을 요청한다

(비고 : 미구현 결정)

이유 : 시스템 밖에서 주로 업무를 처리하는 부서(법무팀) 위주의 유스케이스 - 한정된 개발 시간에서 중요도 낮음)

Description : 구상 처리중인 사건에 대해 손괴자가 요청한 기한까지 구상금을 납부하지 않는 등의 경우 보상팀 직원이 법무팀에게 소송을 부탁하는 액션이다. '손해배상 청구(구상)를 한다' 유스케이스와 include 관계로 형성된 이유는 소송을 하려면 우선 구상 절차가 진행이 되어야하기 때문이다.

- Basic Path

0. 시스템은 보상팀 직원에게 아래와 같은 메뉴들을 보여준다. (E1)

서류 검토하기

보상 여부 결정하기

보상금 책정하기

보상금 지급하기

구상 신청하기

구상 소송 요청하기

1. 보상팀 직원은 '구상소송 요청하기' 버튼을 누른다.

1-1. 시스템은 오늘 날짜가 해당 사건의 구상금 납부기한 날짜와 같거나 그 이후이면서 상태가 '구상 진행중'일 경우 해당 사고의 상태를 '구상소송 필요'로 변경한다.

2. 시스템은 아래와 같이 구상 소송이 필요하거나, 진행중이거나, 완료된 사건들을 나열해서 보여준다.

1. [구상소송 필요] : 사고번호: [구상금 납부기한내에 납부가 안된 사건 id]

2. [구상소송 요청중] 사고번호: [구상 소송이 요청중인 사건 id]

3. [구상소송 진행중] 사고번호: [구상 소송이 진행중인 사건 id]

4. [구상소송 완료] 사고번호: [구상 소송이 완료된 사건의 id]

3.보상팀 직원은 [구상소송 필요] 태그가 붙은 사건을 선택한다.

4.시스템은 자동으로 현재 다른 소송을 맡고있지 않은 랜덤한 법무팀 직원에게

해당 사건을 할당시키고, 아래와 같은 메세지를 전송시킨다.

구상 소송 요청이 들어왔습니다.

사고 번호 :

피보험자 이름:

피보험자 전화번호:

손괴자 이름:

손괴자 전화번호 :

요청 직원 이름:

요청 직원 전화번호:

4-1. 시스템은 해당 사건의 상태를 '구상 소송 요청중'으로 변경한다.

5. 법무팀 직원은 시스템에 로그인한다.

6. 법무팀 직원은 '나의 소송'버튼을 클릭한다.

7. 시스템은 요청된 사건을 아래와 같이 보여준다

[소송 요청됨] 사건번호 : [해당 사건 id]

1. 요청 확인

2. 요청 거절

8. 법무팀 직원은 요청 확인 버튼을 누른다.(A1)

9. 시스템은 '사건 처리 현황' 메뉴에서 해당 사건의 상태를 '구상 소송 진행중'으로 변경한다.

10. 소송이 끝나면 법무팀 직원은 '나의 소송' 메뉴에 들어간다. 시스템은 해당 사건을 아래와 같이 보여준다.

[소송 진행중] 사건번호 : [해당 사건 id]

1. 소송 결과 고지

x. 나가기

11. 법무팀 직원은 '소송 결과 고지' 버튼을 누른다.

12. 시스템은 1. 승소 2.패소 버튼을 보여주고, 법무팀 직원은 '승소' 버튼을 누른다.(A2)

13. 시스템은 아래의 입력란을 보여준다.

승소or패소금(패소금일 경우 음수로 입력하세요) :

소송 비용 :

14. 법무팀 직원은 내용을 기입하고 '확인' 버튼을 누른다.

15. 시스템은 보상팀 직원에게 "사건번호 [해당 사건 id]에 대한 소송이 완료되었습니다." 메시지를 전송한다.

16. 시스템은 소송 비용과 승소금을 해당 사건 정보에 자동적으로 추가한다.

17. 시스템은 해당 사건의 상태를 '구상소송 완료(승소)'로 변경한다.

- A1, 법무팀이 해당 사건의 구상 소송을 거절할 경우

1. 법무팀 직원은 '소송 처리 현황'메뉴를 클릭한다.

6. 법무팀 직원은 해당 사건을 조회하고 '거절' 버튼을 클릭한다.

4. 시스템은 거절 사유 입력란과 "확인" 버튼을 보여준다.

5. 법무팀 직원은 해당 정보를 기입하고 확인 버튼을 누른다.

6. 시스템은 보상팀 직원에게 아래의 메시지를 보낸다.

사고번호 [해당 사고 id]에 대한 구상 소송 요청이 거절되었습니다.

거절 사유 : [입력한 사유]

7. 보상팀 직원은 '사건 처리 현황' 메뉴에서 해당 사건의 상태를 '구상소송 거절'로 변경한다.

- A2, 소송에서 패소했을 경우

1. 법무팀 직원은 해당 사건 항목의 '소송 결과' 란에 '패소' 를 입력한다.

2. 법무팀 직원은 패소 금액과 패소 사유에 대한 소견을 적고 '확인' 버튼을 누른다.

3.[Basic Path] 13~16번이 동일하게 진행된다.

4. 시스템은 해당 사건의 상태를 '구상소송 완료(패소)'로 변경한다.

- E1, 화면이 7초 이상 표시되지 않을 경우

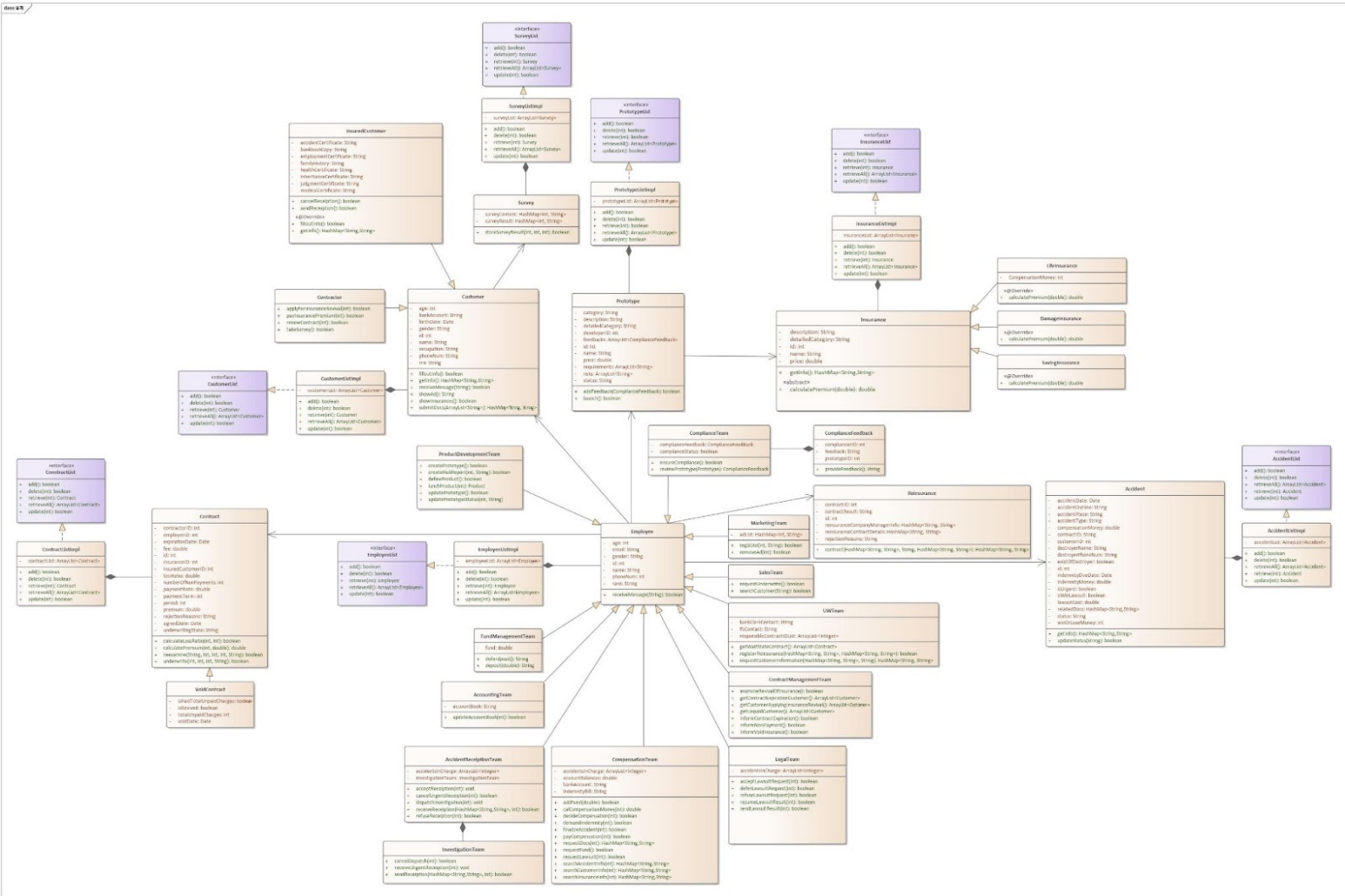
[Error!] 현재 시스템의 성능 저하로 서비스가 원활하게 진행되지 못하고 있습니다. 잠시후 다시 시도해 주세요.

전산팀 사내 유선번호 :047

설계 : Class Diagram

1. Overall Class Diagram

아래는 도출한 유스케이스 다이어그램을 기반으로 설계한 클래스 다이어그램이다.



2. Class Description

Customer

설계자 : 김민수

클래스 설명 : 보험사 시스템을 이용하는 모든 고객이다. 시스템을 이용하는 고객들은 실제로 계약을 맺은 계약자와 실제 보험을 적용받는 피보험자 클래스로 나누어지며 계약자와 피보험자가 공통적으로 가지고 있는 특성들을 지닌 상위 클래스이다.

Attributes

Name	Data Type	Description
age	int	고객의 나이이다.
bankAccount	String	고객의 은행계좌이다.
birthDate	Date	고객의 생일이다.
gender	String	고객의 성별이다.
id	String	고객의 고유 ID로 고객을 구분할 때 사용된다.
name	String	고객의 이름이다.
phoneNum	String	고객의 전화번호이다.
rrn	String	고객의 주민등록번호이다.
occupation	String	고객의 직업이다.

Operations

Name	Parameters	Return Type	Description
filloutInfo		boolean	고객의 정보를 입력하는 메소드이다. 입력하는 값을 해당 클래스의 필드들에 저장한다.
getInfo		HashMap<String, String>	고객의 정보를 반환하는 메소드이다.(이 클래스에 저장된 모든 필드의 이름과 값을 key-value로 매핑하여 반환한다.)
showMessageForCustomer	Customer customer, String message	boolean	시스템이나, 직원이 고객에게 보내는 메시지와 알림을 고객의 화면에 표시하는 메소드이다.
showAd		String	시스템을 통해 보험 광고를 확인하는 메소드이다.
showInsurances		boolean	시스템을 통해 보험 상품들을 확인하는 메소드이다.
submitDocs	ArrayList<String>	HashMap<String, String>	고객이 회원가입을 하거나, 보험에 가입을 할 때 제출해야 하는 서류들을 제출하는 메소드이다. 인자로 받은 ArrayList에는 제출해야 할 서류들의 목록이, 반환값으로 나가는 HashMap에는 제출한 서류의 이름과 그 파일의 값이 key-value 로 매핑되어 반환된다.

Contractor extends Customer

설계자 : 김나연

클래스 설명 : 고객 중에서도 보험을 계약한 사람, 즉 계약자를 말한다. 상위 클래스인 고객을 상속받으며 보험료 납입이나, 재계약, 부활 신청 등 보험 계약자만의 operation을 가진다.

Operations

Name	Parameters	Return Type	Description
applyForInsuranceRevival	int insuranceID	boolean	보험 부활 신청하는 함수. 실효보험에 대해 부활을 신청한다. Parameter로 보험ID를 전달하여, 부활 신청이 성공적으로 처리되면 true, 실패하면 false를 반환.
payInsurancePremium	int insuranceID	boolean	보험료 납입하는 함수. Parameter로 납입 대상 보험ID를 전달하여 보험료 납입이 정상 처리되면 true, 실패하면 false 반환.
renewContract	int insuranceID	boolean	보험 재계약하는 함수. Parameter로 재계약 대상 보험ID를 전달하여 재계약에 성공하면 true, 실패하면 false 반환.
takeSurvey		boolean	보험 가입을 완료한 계약자가 시스템을 통해 설문조사에 참여하는 메소드이다. 계약자가 설문조사 참여에 완료하면 완료 여부를 반환한다.

InsuredCustomer extends Customer

설계자 : 김건우

클래스 설명 : 실제적인 보험의 혜택을 적용받는 피보험자의 클래스이다. 보험이 적용되는 대상이기 때문에 보험을 가입할 때 가입요건에 맞는지, 보상을 진행할 때 사고 증명서 등 추가적인 서류가 요구될 일이 많기에 갖춰야할 정보나 서류가 일반적인 고객에 비해 월등히 많다. 따라서 서류들을 필드로 저장해두어 직원이 서류를 요청 시 바로 제출이 가능하게끔 한다.

Attributes

Name	Data Type	Description
accidentCertificate	String	사고확인서
employmentCertificate	String	재직증명서
healthCertificate	String	건강검진서(종합적인 건강정보)
medicalCertificate	String	진료확인서(사고 등 발생시 어디를 다쳤고, 진료비가 얼마나 나왔는지 등 보상 처리와 같은 특수한 업무에 이용되는 서류)
judgementCertificate	String	법원판결문
familyRelationCertificate	String	가족관계증명서
familyHistory	String	가족력 정보
bankbookCopy	String	통장사본

Operations

Name	Parameters	Return Type	Description
sendReception		boolean	사고를 접수하는 메소드로, 사고 일시, 사고 장소, 사고 유형, 긴급 접수 여부 등의 정보를 기입하는 화면이 표시되며, 기입을 마치면 사고접수팀 직원이 접수를 수신한다.
cancelReception		boolean	접수한 사고를 취소하는 메소드이다. (유스케이스에 명시된 경우에 따라서 취소 수수료가 발생할 수 있다.)

Survey

설계자 : 김민수

클래스 설명 : 마케팅 팀이 시스템 외부적으로 광고를 제작할 때 참고하기 위해 고객들을 대상으로 설문 조사를 할 수 있도록 도와주는 클래스이다. 해당 클래스는 보험 상품에 가입을 완료한 고객들을 대상으로 동작하며 간단한 설문에 대한 답을 할 수 있도록 해준다. 이후 답변들을 리스트로 저장하여 이후 설문에 대한 답변을 확인할 수 있다.

Attributes

Name	Data Type	Description
id	int	설문조사 질문들의 id이다.
surveyQuestionNum	int	설문조사 질문들의 번호이다.
customerChoice	int	고객이 설문조사에서 선택할 번호들이다.
surveyQuestionContent	String	설문조사 질문의 내용이다.
selectedCount	int	해당 질문이 선택된 횟수이다.

Operations

Name	Parameters	Return Type	Description
storeSurveyResult	int, int, int	boolean	설문조사의 결과를 저장하는 메소드로 고객이 설문조사를 완료하면 해당 결과를 리스트에 저장해준다. 이때 설문지 에 대한 답변들은 각 질문마다 정해진 번호를 저장하게 된다. ex) 나이를 선택해주세요 (1) 10~19 (2) 20~29 ...

Insurance

설계자 : 유병규, 김건우

클래스 설명 : 각 세부 보험들의 부모클래스로, 각 보험의 공통적인 attribute와 operation을 갖고 있는 클래스이다. 해당 클래스의 attribute는 모두 접근제어자를 protected로 선언하여 자식 클래스들이 사용할 수 있게 한다.

Attributes

Name	Data Type	Description
description	String	보험에 대한 개요, 보상조건 등 보험에 대해 간략히 설명하는 정보이다.
detailedCategory	String	보험의 세부 카테고리, Prototype 클래스의 category 변수가 LifeInsurance(생명보험), DamageInsurance(손해보험), SavingInsurance(저축보험)을 구분했다면, detailedCategory는 그 category 내에서 세세하게 보험 종류를 나누는 변수이다. 즉, category가 생명보험이었다면 그 안에서 암보험인지, 사망보험인지 분류하는 것이 detailedCategory이다.
id	int	보험을 구분하기 위해 사용하는 식별자이다.
name	String	보험 상품명이다.
price	double	기본적인 보험료이다. 후에 고객과 보험계약이 체결되면 Contract 클래스에서 각 고객의 정보나, 보험의 종류에 맞는 요인들과 복합적으로 계산되어 고객이 내야할 보험료를 구하는데에 쓰인다.

Operations

Name	Parameters	Return Type	Description
getInfo		HashMap<String,String>	해당 보험의 정보들을 반환하는 메소드이다.
<<abstract>> calculatePremium	double paymentRate	double	고객이 내야 할 보험료를 계산하는데 쓰이는 메소드이다. UW팀이 손해율을 분석하여 적절한 요율을 구하고 calculatePremium메소드가 실행되면서 매개변수로 요율이 담긴 채 이 메소드를 호출하는데, 이때 매개변수로 받은 요율(paymentRate)을 보험료를 계산하는데 사용한다.

LifeInsurance, DamageInsurance, SavingInsurance, extends Insurance

설계자 : 유병규, 김건우

클래스 설명 : Insurance를 상속받고 있는 각 세부 보험 유형이다. 각 보험별 로 보험료의 정도와, 그 계산 방식이 다르기에 부모 클래스인 Insurance의 calculateRate 메소드를 오버라이드하여 각 보험의 종류에 맞게 요율을 달리 계산해야 하는것이 Insurance 클래스를 자식클래스로 세분화한 주된 이유이다. LifeInsurance는 생명보험을 말하며 이 보험 카테고리의 세부 종류로는 암보험, 사망보험 등이 있다. 생명보험의 특징은 미리 보상금액이 정해져있다는 것이다(사망시 1억원 지급 등). 따라서 생명보험에만 **compensationMoney**라는 변수를 둔다. DamageInsurance는 손해보험을 말하며, 보험의 세부 종류는 자동차보험, 화재보험, 여행보험, 해상보험 등이 있다. SavingInsurance는 저축보험으로, 연금보험과 일반적인 저축보험 등이 이 카테고리에 속한다. 손해보험은 피해 정도에 따라 보상금이 달라지고, 저축보험은 얼마나 오랜 기간에

걸쳐 얼마만큼의 금액을 저축했느냐에 따라 보상금이 달라지기 때문에 LifeInsurance 클래스와 달리 compensationMoney 변수를 미리 정의해놓지 않는다.

Operations

Name	Parameters	Return Type	Description
<<Override>> calculatePremium	double paymentRate	double	<p>각 보험별 요율과 보험료를 계산하는데에 필요한 요인들을 통해 고객이 내야 할 보험료를 계산하는 메소드이다.(Attribute인 price와 이 메소드에서 구하고자 하는 premium은 다른 개념이다. price는 이 보험의 기본적인 가격이고, calculatePremium은 이 보험의 기본적인 가격에 요율 등 다른 요인들을 모두 합산하여 고객이 내야할 최종 보험료를 구하는 메소드이다.) 이 메소드는 Contract 클래스의 caluatePremium 메소드가 실행될 때 호출된다. 클래스의 변수인 detailedCategory의 종류에 따라 다른 계산법이 적용된다.</p>

Reinsurance

설계자 : 유병규

클래스 설명 : 재보험 계약에 대한 내용을 담고 있는 클래스이다. 재보험은 계약한 보험에 대한 위험부담을 줄이기 위해 해당 보험에 대한 보험을 드는 것으로 보험 회사를 위한 보험이다. 이를 위해 재보험 클래스에는 재보험 계약에 필요한 내용들을 attribute로 계약 체결을 operations로 가지고 있다.

Attributes

Name	Data Type	Description
id	int	재보험을 식별하기 위한 고유 식별 번호다.
contractID	int	재보험을 계약할 Contract의 고유 식별 번호다.
contractReslut	String	재보험 계약 결과를 의미하며, “승인” 혹은 “거절”이 저장된다.
rejectionReasons	String	재보험 계약이 거절된 사유다.
reinsuranceCompanyManagerInfo	HashMap<String, String>	재보험 회사의 담당자 정보로 UWTeam의 registerReinsurance operation을 통해 저장된다.
reinsuranceContractDetailes	HashMap<String, String>	재보험 계약 상세내용으로 UWTeam의 registerReinsurance operation을 통해 저장된다.

Operations

Name	Parameters	Return Type	Description
contract	HashMap<String, String>, String reinsuracneCompanyManagerCon tact, HashMap<String, String>	HashMap<String, String>	재보험 계약을 진행하는 operation이다. 첫번째 파라미터는 계약의 상세내용을 의미하고 두번째 파라미터는 재보험 회사의 담당자 연락처, 세번째 파라미터는 계약을 진행하는 직원의 정보다. 반환되는 값에는 계약 체결 여부, 거절 사유를 담고 있으며 체결되었다면 “승인”, “”이 반환된다. 만약 거절되었다면 “거절”, “거절사유”가 반환된다. 반환된 값에서 계약 체결 여부는 contractResult에, 거절 사유는 rejectionReasons에 저장된다.

ProtoType

설계자 : Utku Gulbardak

클래스 설명 : 상품 개발팀이 보험 상품을 출시하기 전에 만드는 상품 초안이다.

컴플라이언스 팀의 피드백을 받으며 지속적으로 프로토타입의 정보들을 수정하며, 후에
컴플라이언스 팀의 최종 승인을 받아 상품을 출시하면 이곳에 저장되어 있는 정보에
추가적인 정보를 더하여 진정한 Insurance가 만들어진다.

Attributes

Name	Data Type	Description
description	String	This information briefly explains the prototype, such as an overview of its requirements and functions. .
feedbacks	ArrayList <Compliance Feedback>	This is the feedback that comes from compliance feedback class.
id	int	This is an Identifier that identifies the prototype
price	double	This is an estimated price for the product. Price changes due to the insurance category.
requirements	ArrayList<String>	This explains the requirements of the prototype. Requirements can differ depending on insurance type..
risks	ArrayList<String>	This is the list of the possible problems.
status	String	status can be “pending”, “confirmed” or “complete”.
name	String	name of the current prototype that is working on.
category	String	wide category such as “Life Insurance”, “Damage Insurance”, “Saving Insurance”
detailedCategory	String	detailed category within category variable. for example, within “Damage Insurance”category, detailed category can be “car

		insurance", "marine insurance", "fire insurance"
developerID	int	id for who developed this prototype

Operations

Name	Parameters	Return Type	Description
addFeedback	ComplianceFeedback feedback	boolean	method that adds feedback. After running this method, Prototype's feedback field will be updated.
launch		boolean	상품개발팀에서 lunchProduct 메소드를 실행시키면 호출되는 메소드로, 프로토타입에 입력된 정보들을 기반으로 실제 보험상품을 출시하게 된다 (Insurance 클래스의 객체를 생성하고, 그 안에 이 Prototype 클래스의 필드들의 값과 추가적인 정보를 입력하여 완전한 보험상품 생성)

Employee

설계자 : 유병규

클래스 설명 : 각 부서별 직원들의 부모클래스로, 보험 회사에 근무하는 모든 직원들의 기본 정보를 attribute로 갖고 있는 클래스이다.

Attributes

Name	Data Type	Description
id	String	직원을 식별하기 위해 각 직원에게 부여되는 고유 번호이다.
name	String	직원의 이름이다.
gender	String	직원의 성별이며 “남” 혹은 “여”로 저장된다.
email	String	직원이 사용하는 이메일이다.
phoneNum	int	직원의 개인 전화 번호이다.
rank	String	직원의 직급이다.
age	int	직원의 나이이다.

Operations

Name	Parameters	Return Type	Description
showMessageForEmployee	Employee employee, String message	boolean	시스템이나 고객, 다른 직원에 의해 메시지가 발생할 경우 이를 화면에 표시해주는 메소드이다.

ProductDevelopmentTeam extends Employee

설계자 : Utku Gulbardak

클래스 설명 : 제품 개발 팀은 신제품을 정의하고 개발하는 일을 담당합니다. 그들은 컴플라이언스 팀과 긴밀히 협력하고 함께 프로토타입을 만들고 지속적으로 업데이트한 다음 위험 보고서를 작성하고 수익성이 있는지 확인한 다음 제품을 출시합니다.

Operations

Name	Parameters	Return Type	Description
createPrototype		boolean	After definition, the second step is to start creating it. When creating status is yes, the team goes to the next step.
createRiskReport	int, String	boolean	After the latest updates, the development team creates a risk report. When the prototype is ready, its status turns yes.
defineProduct		boolean	When the definition of the product is ready, its status is yes and the team goes to the next step.
launchProduct		Product	The last step of the development team. They execute the program and launch the product.
updatePrototype		boolean	After discussing with the compliance team, the development team updates the prototype and its status turns yes.
updatePrototypeStatus	int,String		When all the parameters are ready, the development team updates prototype Status that it is ready to launch.

ComplianceTeam extends Employee

설계자 : Utku Gulbardak

클래스 설명 : 프로토타입이 개발되고 실제 보험으로 출시되는 과정에서 피드백과 출시 승인을 해주는 컴플라이언스 팀의 클래스입니다.

Attributes

Name	Data Type	Description
complianceStatus	boolean	status of the ongoing compliance report.
complianceFeedback	ComplianceFeedBack	ComplianceFeedBack 클래스는 이 클래스에 종속적이기 때문에 설계원칙에 의거해 ComplianceFeedBack 타입의 필드를 설정해놓고 이를 관리한다.

Operations

Name	Parameters	Return Type	Description
ensureCompliance		boolean	This method is the compliance team ensuring if the product is compliant with customers requirements.
reviewPrototype	Prototype	Compliance Feedback	This method comes from the compliance feedback class and extends to prototype after compliance team's confirmation.

Compliance Feedback

설계자 : Utku Gulbardak

클래스 설명 : 컴플라이언스 팀이 남기는 피드백으로, 어느 프로토타입에 대해 어느 컴플라이언스 직원이 어떤 피드백을 남겼는지를 나타내고 있습니다.

Attributes

Name	Data Type	Description
complianceID	int	This represents the unique ID number of the compliance team member..
feedback	String	This represents the comments and ideas of the compliance team member.
prototypeID	int	This represents the unique ID of the prototype being worked on.

Operations

Name	Parameters	Return Type	Description
provideFeedback		String	This method represents providing feedback from compliance team members. It's included in the compliance team.

MarketingTeam extends Employee

설계자 : 김민수

클래스 설명 : 보험사에서 마케팅, 특히 보험 상품에 대한 광고를 관리하는 부서이다. 이 클래스는 보험 가입을 완료한 고객을 대상으로 진행한 설문조사의 결과를 바탕으로 시스템 외부에서 완성한 광고를 해당 시스템에 등록하거나, 삭제하는 클래스이다.

Attributes

Name	Data Type	Description
asName	String	광고할 상품의 이름이다.
adDescription	String	광고할 상품에 대한 설명이다.

Operations

Name	Parameters	Return Type	Description
registAd	String, String	boolean	광고를 등록하는 메소드이며, 광고의 이름과 광고 내용을 입력하여 광고를 등록하고 등록 여부를 표시해준다.

SalesTeam extends Employee

설계자 : 김민수

클래스 설명 : 보험사에서 상품을 고객에게 추천하거나, 고객과의 상담을 통해 상품 판매에 영향을 미치는 영업 부서이다. 이 클래스는 고객에게 보험 상품을 추천해주거나, 고객이 보험 가입을 신청하면 해당 신청에 대한 승인 받기 위해 U/W 부서에게 승인을 요청하고 끝으로 고객의 보험 가입을 완료해주는 클래스이다.

Operations

Name	Parameters	Return Type	Description
requestUnderWrite		boolean	고객이 보험상품 가입을 신청하면 영업팀 직원이 후에 고객과의 계약을 진행하기 위해 UW팀 직원에게 인수심사를 요청하는 메소드이다. 이 메소드가 실행되면 Contract 클래스의 객체가 생성되고, 생성과 동시에 신청한 고객의 id와 신청한 보험상품의 id, 그리고 해당 계약을 체결하는 영업팀 직원의 id가 기록되며, 해당 객체의 status 는 'wait(인수심사 대기중)' 상태가 된다.
searchCustomer	String	boolean	고객에게 알맞는 보험 상품을 추천하기 위해 고객의 정보를 찾는 메소드로 원하는 조건을 입력하면 해당 조건에 부합하는 고객이 있는지 확인해준다. 이후 고객의 세부 정보는 DB를 통해 확인하게 된다.

UWTeam extends Employee

설계자 : 유병규

클래스 설명 : U/W 부서에서 근무하는 직원을 의미하는 클래스이다. U/W 부서에서는 보험 인수 심사와 재보험 업무 처리를 담당하는 부서이다. 인수 심사와 재보험 업무 처리 과정에서 발생할 손해율을 분석하는 업무 또한 담당하며 보험 인수를 위한 고객의 신용 정보 조회, 금융감독원의 심사가 이뤄진다.

Attributes

Name	Data Type	Description
bankClerkContact	String	은행원 연락처이다. U/W 부서 직원은 인수 심사 과정에서 고객의 신용정보를 수집해야 하기 때문에 이를 위한 은행원 연락처를 보유하고 있어야 한다.
ffsContact	String	금융감독원 연락처이다. U/W 부서 직원은 인수 심사 과정에서 금융감독원의 검사(심사)를 요청해야한다. 그렇기 때문에 금융감독원의 연락처를 보유하고 있어야 한다.
responsibleContractIDList	ArrayList<Integer>	해당 직원이 처리한 계약의 고유 식별 번호를 담고 있는 리스트이다. U/W 부서에는 많은 직원들이 많은 계약을 심사하기 때문에 어떤 직원이 어떤 계약의 인수 심사에 참여하였는 지를 저장해야 한다.

Operations

Name	Parameters	Return Type	Description
getWaitStateContract		ArrayList<Contract>	U/W 부서 직원이 인수 심사를 하기 위해서 아직 인수 심사를 하지 않은 계약들을 리스트로 불러오는 operation이다. 이 operation를 호출하면 Contract 클래스의 underwritingState attribute가 “대기” 상태인 Contract를 리스트에 담아 반환해준다.
registerReinsurance	HashMap<String, String>, HashMap<String, String>	boolean	재보험 계약을 등록하는 operation이다. 여기서 등록은 계약 체결 되기 전 계약의 생성을 의미한다. 첫번째 파라미터는 재보험 계약의 상세내용을 담고 있고 두번째 파라미터는 재보험 계약을 맺을 재보험 회사의 정보, 해당 회사의 담당직원의 내용을 담고 있다. 해당 operation이 실행되면 이 두 파라미터를 통해 Reinsurance의 attribute를 설정하고 객체를 생성한다. 계약 체결은

			Reinsurance의 contract operation으로 처리된다.
requestCustomerInformation	HashMap<String, String>, String baankClerkContac	HashMap<String, String>	고객의 정보를 요청하는 operation이다. 첫번째 파라미터는 요청할 고객의 기본정보(주민등록번호, 이름)를 담고 있고 두번째 파라미터는 bankClerkContact로 요청할 은행원의 연락처를 담고 있다. 해당 operation이 실행되면 두 파라미터를 통해 은행원에게 요청할 고객의 기본정보를 전달한 후 은행원으로부터 고객의 정보를 전달 받는다.

ContractManagementTeam extends Employee

설계자 : 김나연

클래스 설명 : 계약관리팀에 근무하는 직원 클래스. 계약관리팀은 보험료 미납, 만기 계약, 재계약, 실효 계약, 보험 부활 등에 대한 업무를 다룬다. 보험 계약에 변동사항이 생기거나 생길 예정일 경우, 고객에게 안내하는 업무도 포함된다.

Operations

Name	Parameters	Return Type	Description
examineRevivalOfInsurane		boolean	보험 부활 심사하는 함수. 부활 심사 결과를 리턴한다. 심사 결과, 승인일 경우 true, 거절일 경우 false 반환.
getContractExpirationCustomer		ArrayList<Customer>	계약 만기 임박 고객을 조회하는 함수. 계약 만기일이 30일 이하로 남을 경우에만 조회된다.
getCustomerApplyingInsuranceRevival		ArrayList<Customer>	보험 부활을 신청한 고객을 조회하는 함수. 고객이 총 미납료를 모두 지불하고, 대상 보험이 실효

			보험일 때만 부활 심사가 진행된다.
getUnpaidCustomer		ArrayList<Customer>	보험료 미납 고객을 조회하는 함수. 미납횟수가 3회 이상일 경우 실효 계약으로 처리한다.
informContractExpiration		boolean	계약 만기 임박 혹은 계약 만기를 고객에게 안내하는 함수. 재계약 여부를 묻는 메시지를 포함한다. 정상적으로 문자(혹은 전화)가 전송되면 true, 실패 시 false 반환.
informNonPayment		boolean	보험료 미납 사실을 고객에게 안내하는 함수. 미납횟수가 3회 이상이 되면 실효 보험으로 처리된다는 메시지를 포함한다. 정상적으로 문자가 전송되면 true, 실패 시 false 반환.
informVoidInsurance		boolean	보험이 실효됨을 고객에게 안내하는 함수. 정상적으로 문자가 전송되면 true, 실패 시 false 반환.

AccidentReceptionTeam extends Employee

설계자 : 김건우

클래스 설명 : 사고접수팀 직원의 클래스이다.

Attributes

Name	Data Type	Description
investigationTeam	InvestigationTeam	현장출동 팀이다. 사고접수팀에서 현장조사팀을 관리하기 때문에 이 클래스가 생성되면 해당 필드에 현장조사팀을 생성하여 저장해놓는다.
accidentsIncharge	ArrayList<Integer>	직원이 접수받고있는 사건id 들의 리스트이다.

Operations

Name	Parameters	Return Type	Description
receiveReception	String customerID, HashMap<String,String>	boolean	InsuredCustomer(피보험자)클래스의 사고를 접수하는 sendReception

	accidentInfo		메소드가 실행되면 호출되는 메소드로, 고객id와 사고 정보가 담긴 HashMap을 매개변수로 받아 사고 접수팀 직원에게 사고접수에 대한 알림과 사고 정보를 표시한다. 또한 이 메소드가 실행되면서 Accident 클래스의 객체를 생성하여 자동으로 accidentID를 할당하고, 해당 메소드에서 인자로 받은 정보들을 Accident의 필드들에 기록시킨다.(접수시에는 모든 정보가 함께 기록되지 않으니 접수시 고객측에서 입력한 정보들만 매핑이 된다. 매핑이 안된 나머지는 null 또는 0). 또한 이 클래스의 accidentsIncharge 리스트에 accidentID를 추가시킨다.
refuseReception	int accidentID	boolean	receiveReception 메소드가 실행되고 사고 접수팀 직원이 접수 거절을 선택하면 실행되는 메소드이다. accidentD 를 인자로 받아 해당 id와 일치하는 고객클래스의 객체의 receiveMessage 메소드를 호출하여 거절되었다는 메시지를 전달한다. 또한, 인자로 받은 accidentID에 해당하는 Accident의 status를 '접수 거절'로 설정한다.
acceptReception	int accidentID	boolean	receiveReception 메소드가 실행되고 사고 접수팀 직원이 접수 승인을 선택하면 실행되는 메소드이다. 해당 메소드가 종료되면 Accident 클래스의 해당 사건 객체의 status가 '접수 완료'로 설정된다. 접수된 사고가 긴급 접수일경우(매개변수로 받은 해당 사건의 accidentID를 가진 Accident 클래스 객체의 isUrgent 필드가 true일경우) dispatchInvestigation 메소드를

			호출한다.
--	--	--	-------

InvestigationTeam extends Employee

설계자 : 김건우

클래스 설명 : 현장조사팀 직원 클래스이다. 긴급 접수가 들어오면 이를 할당받아 출동한다.

Attributes

Name	Data Type	Description
isDispatching	boolean	출동을 한 상태인지 아닌지를 구분하는 변수이다. true일 경우 이미 출동을 나가있으므로 다른 현장조사팀 직원을 선정해야 한다
accidentID	Integer	해당 현장조사팀 직원에게 할당된 긴급 접수 사건번호이다. int가 아닌 참조형 Integer로 선언된 이유는 긴급 접수가 할당이 안될 경우 NULL로 설정해두기 위함이다.

Operations

Name	Parameters	Return Type	Description
receiveUrgentReception	int accidentID	boolean	고객의 정보와 사고 정보를 현장조사팀에게 보여준다. 현장 조사팀이 확인 버튼을 누르면 고객에게 현장조사팀이 출동했다는 메시지를 전송하고(Customer 클래스의 receiveMessage 호출), 이 클래스의 isDispatching의 값을 true로 설정한다.

sendReception	HashMap<String,String> accidentInfo	boolean	현장조사팀이 출동하여 현장을 탐사한 후, 고객 대신에 자세한 사고 정보를 적어 사고를 접수하는 메소드이다.
---------------	----------------------------------------	---------	-------------------------------------------------------------

CompensationTeam extends Employee

설계자 : 김건우

클래스 설명 : 보상팀 직원 클래스로, 보상에 대한 전반적인 절차를 수행하는 변수와 메소드를 저장하고 있다. AccidentReceptionTeam(사고접수팀 직원) 클래스에서 사고접수를 수락하여 Accident 클래스를 만들고 기본적인 정보들을 기입했다면, 이 클래스에서 각 보상 절차에 관련된 메소드가 동작하면서 Accident 클래스의 추가적으로 정보를 추가하고, 수정하는 방식 위주로 동작한다.

Attributes

Name	Data Type	Description
accountBalances	double	보상팀 직원의 보상 계좌에 들어있는 잔액이다.
bankAccount	String	보상팀 직원의 보상 계좌번호이다.
indemnityBill	String	구상 신청을 할 때 손괴자에게 발송할 구상금 청구서이다.

Operations

Name	Parameters	Return Type	Description
searchAccidentInfo	int accidentID	HashMap<String,String>	accidentID를 통해 보상 처리중인 사건의 정보를 조회하는 메소드이다.
searchCustomerInfo	int accidentID	HashMap<String,String>	accidentID를 통해 해당 사건의 고객 정보를 조회하는 메소드이다.
searchInsuranceInfo	int accidentID	HashMap<String,String>	accidentID를 통해 해당 사건의 고객이 계약한 보험의 정보를 조회하는

			메소드이다.
requestDocs	int accidentID	HashMap<String,String>	보험금 지급 여부를 결정하기에 앞서 고객에게 추가적인 서류를 요청하는 작업이다. 해당 메소드를 실행하면 InsuredCustomer 클래스 객체의 피보험자는 submitDocs 메소드를 통해 서류를 제출한다. 이 메소드가 실행될 때는 Accident의 status를 '서류 요청 중'으로, 피보험자가 서류 제출을 하고 나면 '정보 검토 중'으로 변경한다. 서류 제출이 완료되면 Accident의 realtedDocs에 제출한 서류들이 추가된다.
decideCompensation	int accidentID	boolean	보상금 지급여부를 결정하는 메소드로, 해당 메소드가 완료되면 이 메소드의 매개변수인 accidentID에 해당하는 Accident 클래스 객체의 status가 '보상금 지급 승인', '보상금 지급 거절'등으로 바뀐다.
calCompensationMoney	int accidentID	double	보상금을 책정하는 메소드로, 이 메소드가 실행되었을 때 Accident의 status를 '보상금 책정 중'으로, 최종적으로 금액과 소견을 기입하여 보상금 책정이 완료되어 메소드가 종료되면 Accident의 status를 '보상금 책정 완료'로 변경하고, Accident의 CompensationMoney 의 값을 책정한 보상금으로 설정한다.
payCompensation	int accidentID	boolean	보상금을 지급하는 메소드로, 보상금을 지급하는 과정에서 accountBalance 필드의 값에서 보상한 지급금만큼 차감된다. 이 메소드가 성공적으로 종료되면 Accident의 status가 '보상금 지급 완료'로 변경된다.

requestFund		boolean	자금운용팀에게 보상금으로 지급할 예산을 요청하는 메소드이다.
addFund	double fund	boolean	자금운용팀에게 지급받은 예산을 보상팀 직원의 자금(accountBalances)에 더하는 메소드이다.
demandIndemnity	int accidentID	boolean	해당 사건의 손괴자에게 구상요청을 하는 메소드이다. 이 메소드를 실행시키면 Accident의 status가 '구상 처리 중'으로 바뀌며, 보상팀 직원에게 손괴자의 이름과 연락처가 표시된다. (앞서 유스케이스 시나리오에서도 언급했듯 손괴자는 보험사 시스템의 고객이 아니기에 직접 구상금 입금을 확인해야 하고, 정상적으로 입금시 Accident의 status또한 직접 '구상 완료'로 바뀌줘야 한다.)
requestLawsuit	int accidentID	boolean	법무팀 직원에게 구상소송을 요청하는 메소드이다. 이 메소드를 실행시키면 Accident의 status가 '구상 소송 요청 중'으로 변경된다.
finalizeAccident	int accidentID	boolean	보상 처리를 마무리짓는 메소드로, 여지까지 각 보상 절차를 진행하면서 발생한 지출과 수익금을 시스템이 표시해주고, 이 정보를 회계팀에게 넘겨준다. 이 메소드가 종료되면서 인자로 받은 accidentID의 객체의 status가 '종결'로 변경된다.

FundManagementTeam extends Employee

설계자 : 김건우

클래스 설명 : 회사의 자금을 운용하면서 예산 분배, 각 부서나 직원에 업무금을 할당해주는 자금운용팀 직원 클래스이다.

Attributes

Name	Data Type	Description
fund	double	예산으로 즉시 지급이 가능한 자금이다.

Operations

Name	Parameters	Return Type	Description
deposit	double	String	CompensationTeam 클래스의 requestFund 메소드가 실행되면 호출되는 메소드로, 자금 지원 (입금)을 요청받아 이에 입금해주는 메소드이다. 입금이 되면 이 클래스의 필드인 fund 값에서 입금한 금액만큼 차감된다. (반환값에는 입금 완료 등의 메시지가 기록된다.)
deferDeposit		String	입금을 보류할때 사용되는 메소드이다. 보류 사유 등의 정보가 반환된다.

LegalTeam extends Employee

설계자 : 김건우

클래스 설명 : 법률 업무를 담당하는 법무팀 직원의 클래스로, 보상 절차에서 구상 신청을 할 때 손괴자가 이에 응하지 않아 보상팀 직원이 소송을 맡길 때, 이를 확인하고 소송결과를 알려준다.

Attributes

Name	Data Type	Description
accidentsInCharge	ArrayList<Integer>	소송 요청을 받은 사건들의 id 리스트들이다.

Operations

Name	Parameters	Return Type	Description
acceptLawsuitRequest	int accidentID	boolean	구상 소송 요청을 접수받는 메소드로, CompensationTeam의 requestLawsuit 메소드 이후에 동작하는 메소드이다. 법무팀 직원이 이 메소드에서 '확인'버튼을 누르면 Accident의 Status가 '구상 소송 진행 중'으로 바뀐다.
deferLawsuitRequest	int accidentID	boolean	법무팀 직원이 구상 소송을 보류할 때 작동하는 메소드로 Accident의 Status가 '구상 소송 보류 중'으로 변경된다.
resumeLawsuitRequest	int accidentID	boolean	법무팀 직원이 보류시켰던 소송 을 다시 재개할때 작동하는 메소드로 Accident의 Status가'구상 소송 진행 중'으로 변경된다.
refuseLawsuitRequest	int accdientID	boolean	법무팀 직원이 구상 소송을 거절할 때 작동하는 메소드이다. 해당 메소드는 Accident의 Status를 자동적으로

			변경하지 않고 보상팀 직원에게 거절되었다는 알림만 전하기 때문에 보상팀 직원이 이를 확인하고 Accident의 Status를'구상 완료'로 직접 변경해야 한다.
sendLawsuitResult	int accidentID	boolean	구상 소송을 마치고 그 결과를 전달하는 메소드로, 보상팀 직원은 이 메소드 내에서 소송 결과 소송을 하면서 들었던 지출과 수익을 입력하면 인자로 받은 accidentID에 해당하는 Accodident 클래스의 객체인 해당 사건에 승 패소 여부(isWinLawsuit 필드)와 금액 정보들(lawsuitCost, winOrLoseMoney)의 값을 세팅한다. 마찬가지로 해당 메소드는 Accident의 Status를 자동적으로 변경하지 않고 보상팀 직원에게 소송 결과와 내용만 전달하기에, 이후 보상팀 직원이 이를 확인하고 Accident의 Status를'구상 완료'로 직접 변경해야 한다.

AccountingTeam extends Employee

설계자 : 김건우

클래스 설명 : 회계 정보를 관리하는 회계팀 직원의 클래스이며, 보상 절차가 완료되면,
해당 사건을 통한 지출금과 수익금을 기록하는 역할을 한다. (자금운용팀, 법무팀,
회계팀은 우리에게 소프트웨어 개발을 의뢰한 보험사에서 각 부서별로 별도의 시스템을
사용하고, 필수적인 요구사항에 해당 부서들의 내용은 존재하지 않거나 최소한의 기능만
요구하고 있기에 보상 절차와 관련된 기능과 구조만 설계했다.)

Attributes

Name	Data Type	Description
accountBook	String	대변과 차변으로 구성되어있는 회계정보를 간단히 기록하는 텍스트이다.

Operations

Name	Parameters	Return Type	Description
updateAccountBook	accidentID	boolean	Compensation 팀이 모든 보상 절차를 완료하고 finalizeAccident 메소드를 실행시키면 호출되는 메소드로, Accident 클래스로부터 해당 사건의 지출금/수익금 정보를 가져와 accountBook 필드에 업데이트한다.

Accident

설계자 : 김건우

클래스 설명 : 사고 접수부터 보상 및 사건 종결(보상 절차 모두 완료)까지 보상 절차에 전반적인 진행도와 보상에 필요한 정보를 가지고 있는 일종의 Data Set의 성향을 띄는 클래스이다. 사고접수팀(AccidentReceptionTeam), 보상처리팀(CompensationTeam), 법무팀(LegalTeam), 회계팀(AccountingTeam) 등 보상에 관련된 직원들이 데이터를 조회하고, 필드의 값들을 수정하는 등의 접근이 잦은 클래스이다. 사고접수팀에서 receiveReception 메소드를 통해 사고 접수를 받을 때 해당 클래스의 객체가 생성된다.

Attributes

Name	Data Type	Description
id	int	해당 사건을 구분하는 식별자이다. 사고가 접수되는 시점에 자동생성된다.
customerID	String	사고를 접수한 고객의 ID이다.
receptionEmployeeID	String	해당 사고를 접수받은 직원의 ID이다.
compensationEmployeeID	String	해당 사고의 보상처리를 맡은 직원의 ID이다.
isUrgent	boolean	긴급접수인지를 구별하는 필드이다. 고객이 긴급접수를 할 경우 사고접수팀에서 해당 접수를 받으면서 객체를 생성할때 이 필드의 값을 true로 설정한다.
accidentDate	Date	사고가 발생한 일시이다.
accidentPlace	String	사고가 일어난 장소이다.
accidentType	String	사고의 유형이다(자동차 사고, 암 발병)
accidentOutline	String	사고에 대한 전반적인 개요이다.(접수 시 작성하는 사고 개요)
existOfDestroyer	boolean	해당 사건의 손괴자 여부이다. 존재 시 true, 없을 시 false로 설정된다.
destroyerName	String	손괴자의 이름이다.(손괴자가 없는 사건의 경우 Null)
destroyerPhoneNum	String	손괴자의 연락처이다.(마찬가지로 없는 사건의 경우 Null)
status	String	현재 사건의 처리가 어디까지 진행되었는지 나타내는 상태 정보이다. 각 보상 처리 관련 업무의 단계를 밟아갈 때마다 이 status의 값이 수정된다.
medicalBill	String	해당 사건에 대한 보상금 책정에 참고될 진료비 계산서다.

damageBill	String	해당 사건에 대한 보상금 책정에 참고될 손해 계산서다.
compensationMoney	double	피보험자에게 지급할 보상금이다.
indemnityDueDate	Date	손괴자에게 받아야 할 구상금이 언제까지 입금되어야 한다는 만료일이다.
indemnityMoney	double	손괴자가 보상해야 할 구상금 액수이다.
isWinLawsuit	Boolean (boolean이 아닌 참조형인 Boolean으로 선언한 이유는 승소 : true, 패소: false, 구상 소송이 없거나 아직 판결이 안난 사건의 경우 null이어야 하기 때문이다.)	구상 소송을 진행한 후 그 소송 결과이다.
lawsuitCost	double	구상 소송을 치르면서 발생한 비용이다.(소송 준비금, 변호사 선임비 등)
winOrLoseMoney	double	구상 소송을 치르면서 판결에 따라 지불하거나 얻은 승소금 혹은 패소금이다. 승소했을 경우 양수, 패소했을 경우 음수가 저장된다.

Operations

Name	Parameters	Return Type	Description
getInfo		HashMap<String,String>	이 클래스의 모든 필드들의 이름(key)과 값(value)을 HashMap 구조로 반환하는 정보 조회용 메소드이다.

Contract

설계자 : 유병규

클래스 설명 : 고객과의 보험 계약에 관한 클래스이다. 보험 계약에 필요한 정보들을 attribute로 가지고 있으며 해당 계약의 처리는 operations로 진행된다. 영업 부서 직원에 의해 Contract 객체가 생성되며 계약 처리에 관한 operations는 U/W 부서 직원에 의해 실행된다.

Attributes

Name	Data Type	Description
id	int	계약의 고유 식별 번호다.
insuranceID	int	계약 대상 보험의 고유 식별 번호다.
employeeID	int	계약을 등록한 영업 부서 직원의 고유 식별 번호다.
contractorID	int	계약을 진행하는(보험료를 납부하는) 고객의 고유 식별 번호다.
insuredCustomerID	int	보험이 적용대상이 되는 피보험자 고객의 고유 식별 번호다.
expirationDate	Date	계약이 만료되는 날짜다.
paymentTerm	int	보험료를 납부해야 하는 주기이다(예를 들어, 저축성 보험의 경우 한달마다 꾸준히 보험료를 내서 저축하는 경우가 대부분이니 값이 30, 연마다 보험료를 납부하는 생명보험의 경우 값이 365가 된다.)
paymentRate	double	계약에 적용되는 요율이다.
lossRatio	double	계약이 체결될 시 발생할 손해율이며 calculateLossRatio operation을 통해 값이 저장된다.
premium	double	고객이 납부해야할 보험료다. calculatePremium operation을 통해 값이 저장된다.
period	int	계약이 적용되는 기간이다.

signedDate	Date	계약을 최종 체결한 날짜다.
numberOfNonPayments	Int	보험료 미납 횟수. 3회 이상일 경우 계약관리팀이 실효 계약으로 취급한다.
underwritingState	String	인수심사 상태를 말하며 “대기”, “승인”, “재심사 가능”, “재심사 거절”이 저장된다.
rejectionReasons	String	인수(재)심사가 거절되었을 시 거절 사유를 저장한다.
fee	double	피보험자가 긴급 사고접수 서비스를 이용하는 등의 행위로 추가적으로 발생한 수수료를 저장한다.

Operations

Name	Parameters	Return Type	Description
calculateLossRatio	int insuranceID, int cutomerID	boolean	고객의 정보(직업, 재산, 신용, 가족력)와 해당 보험의 기본 금액, 납부기간을 통해 해당 계약이 체결되었을 시 회사가 겪을 손해율을 계산하는 operation이다. 손해율 계산에 성공하였다면 true를, 실패하였다면 false를 반환하며 true일 경우 계산된 손해율을 lossRatio attribute에 저장한다.
calculatePremium	int insuranceID, double paymentRate	double	요율과 보험의 기본 금액을 통해 고객이 납부해야할 최종 금액을 산출한다. 이때 반환 값이 premium attribute에 저장된다.
reexamine	String underwritingState, int employeeID, int insuranceID, int customerID, String ffsContact	boolean	underwritingState가 “재심사 가능”일때만 동작하는 operation으로 underwritingState 파라미터를 제외한 4개의 파라미터를 underwriting operation의 파리미터로 주어 underwrite operation을 호출한다. 이때 underwrite의 결과를 반환해준다. 만약

			underwrite의 결과가 false일 경우 underwritingState를 “재심사 거절”로 변경하고 false를 반환한다.
underwrite	int employeeID, int insuranceID, int, customerID String ffsContact	boolean	인수 심사를 수행하는 operation이다. 파라미터로 제공된 insuranceID와 customerID를 통해 보험 정보와 고객 정보를 확인하고 자체적으로 심사를 진행한다. 그리고 ffsContact를 통해 금융감독원에게 계약 승인 여부를 받는다. 인수 심사 결과 승인이라면 underwriting을 “승인”으로 변경하고 true를, 거절이라면 “재심사 가능”로 변경하고 false를 반환한다. 만약 false일 경우는 rejectionReasons에 거절 사유를 저장한다.

VoidContract extends Contract

설계자 : 김나연

클래스 설명 : 실효 계약 클래스. 고객이 보험을 가입하고, 보험료 미납 횟수가 3회 이상이 되면 해당 보험이 실효 처리된다. 실효 계약은 말 그대로 효력을 상실한 보험이므로, 사고가 나도 보험금을 지급받지 못하는 등 여러가지 제한이 생긴다.

Attributes

Name	Data Type	Description
isPaidTotalUnpaidCharges	boolean	총 미납료를 납입했는지 여부 (부활 심사 조건에 포함됨)
isRevived	boolean	부활 승인되었는지 여부
totalUnpaidCharges	int	총 미납료
voidDate	Date	실효일

맺으며

팀원 별 소감

아래는 분석과 설계를 마치며 팀원들이 느꼈던 소감이다.

- 김건우 : 평소에는 개발 프로젝트를 진행할 때 설계를 하지 않고 바로 개발부터 들어가거나, 간단한 설계만 하고 개발을 하곤 했는데 이번 기회를 통해 체계적으로 설계하는 방법을 익힐 수 있었다. 특히, 실제 기업의 제안서를 꼼꼼히 검토하고 분석하여 팀원들과 커뮤니케이션 하며 무엇이 더 올바른 설계원칙에 걸맞는지 끊임없이 토의하고 고민해보면서 설계 실력과 논리적인 사고 능력이 많이 상승할 수 있었던 것 같다. 이에 더해, 소프트웨어 엔지니어에게 개발 지식만큼 중요한 것이 자신이 개발하는 제품의 도메인 영역에 대한 지식의 습득력과 깊이임을 깨달을 수 있는 유익한 경험이었다.
- 김나연 : 팀프로젝트1 수업과 같이 진행하다보니, 애자일 프로세스와 UseCase를 통해 요구사항을 분석하고 설계하는 개발과정의 차이점과 장단점을 확 느낄 수 있었다. 확실히 애자일이 덜 귀찮다라는 생각이 든다. 일단 환경 세팅만 하고 바로 개발을 시작할 수 있으며, 거의 실시간으로 고객의 요구사항을 반영할 수 있다는 장점이 좋았다. 다만 분석과 설계가 제대로 잡혀있지 않기 때문에, 팀원간의 협업에서 소프트웨어 구조, 구현의 일관성을 유지하기가 어려웠고, 사소한 것에서 생각의 차이가 발생하는 부분이 있어서 아쉬웠었다. 완성도를 높이려면 서로 이런 점들을 맞춰야 해서 또 시간이 걸리더라. 바로 이런 점에서 UseCase 시나리오, Class Diagram을 통해 요구사항 분석, 설계를 진행하는 것이구나 생각했다. 결국 공통적으로 추구하는 것은 고객의 요구사항에 맞추는 것이다. 고객의 요구사항이 매우 구체적이고 변동될 가능성이 낮다면 오히려 분석과 설계를 모두 마치고 개발을 시작하는 것이 유리할 수도 있겠다는 생각이 들었다.
- 김민수 : 팀을 짜고 팀원들과 프로젝트를 진행하는 것이 처음이다 보니, 상당히 어려움을 느꼈다. 특히 나의 생각과 다른 팀원들의 생각이 전혀 다른 상황에서도 서로 같은 생각을 하는 듯한 느낌이 들 때 답답함을 느끼기도 하였다. 하지만 이번 프로젝트의 설계를 하면서 나의 생각을 표현하는 방법을 알게된 것도 같아 좋았던 것 같다. 내 생각을 어떻게 상대방에게 정확히 전달할 수 있는지, 그리고 정확히 전달하는 것의 중요성 역시 알 수 있었던 흥미로운 경험이었다. 또한 소프트웨어를 설계를 하는 일이 없어 설계를 하면서 헛갈리는 상황이 많아 상당히 힘들었는데, 팀원들과 토의하면서 최선의 결과를 내려 노력한 경험들이 유익했던 것 같다.

- 유병규 : 개인 프로젝트를 진행하는 것과 고객의 프로젝트를 진행하는 것은 전혀 다르다는 것을 알 수 있었다. 자신의 생각이 아닌 타인이 생각하는 결과물을 만드는 과정에서 코딩뿐만이 아니라 프로젝트에 필요한 도메인 지식의 이해, 요구사항 분석, 시스템 설계, 설계도와 같은 도식화 방법 등이 매우 중요한 요소임을 깨닫고 경험해 볼 수 있는 좋은 기회였다.
- Utku Gulbardak : I realize that analysis and design is the hardest and also the most important part of a good project. Creating something in my mind and implementing it with my team members as well as with a client and making continuous updates until perfecting a project requires so much attention and patience.
분석과 설계가 좋은 프로젝트의 가장 어려운 부분이자 가장 중요한 부분이라는 것을 깨달았습니다. 내 마음 속에 무언가를 만들고 그것을 클라이언트뿐만 아니라 팀원들과 함께 구현하고 프로젝트를 완성할 때까지 지속적으로 업데이트하는 것은 많은 관심과 인내가 필요합니다.

앞으로의 방향성

iGO 팀은 많은 유스케이스의 분량을 뽑아내기 보다는 보험사 시스템의 필수적인 세부 도메인들의 기능 요구사항에 기반하여 정확성과 완전성에 중점을 두었고, 이를 고객에게 충분히 설명할 수 있도록 목표하며 설계를 진행했다. 우리는 이 설계를 가지고 최소한의 변경만을 통해 실제 소프트웨어를 완성해내는 것에 최우선 목표를 두고 있다. 변동사항이 존재하지 않는 완벽한 설계는 존재하지 않겠지만, 그럼에도 그러한 방향을 지향하며 고객의 요구사항에 초점을 맞춰야한다는 방향성을 팀원 모두가 꾸준히 리마인드하며 구현 단계에서 제품이 고객을 위한 소프트웨어에서 개발자가 구현하기 편한 소프트웨어로 변형되지 않도록 노력할 것이며, 설계와 구현 사이의 모순을 최소화하는데 집중할 것이다.