

2025년 전남대학교 소프트웨어중심대학사업 소·중·대 산학협력프로젝트(캡스톤디자인) 결과보고서

프로젝트명	오토피봇 (Auto Fee Bot : 회비 자동 관리 시스템)						
Github url 주소	https://github.com/kchan1057/autopivot-fee-management-BE https://github.com/kchan1057/autopivot-fee-management-FE						
팀 명	정성은 다해요			과제수행기간	2025. 9. 24. ~ 12. 19.		
지도교수	학 과	컴퓨터정보통신공학과		성 명	유선용		
프로젝트 수행인원 (※팀장은 첫줄에 기입)	이 름	학과(부·복수전공)	학년	학번	연락처(HP)	E-Mail	
	팀장	정주희	컴퓨터정보통신공학과	4	214732	010-6788-7585	7585sy@naver.com
	팀원	김성찬	컴퓨터정보통신공학과	3	21803	010-4690-8377	awdr1057@naver.com
참여 기업	기업명	멘토명	직위	연락처(HP)	E-Mail		
	(주)글로즈	임성진	사장	010-3646-0999	-		

위와 같이 2025년 전남대학교 소프트웨어중심대학사업
산학협력프로젝트 지원 프로그램 결과보고서를 제출합니다.

2025년 12월 19일

신청자명(대표학생) : 정주희
지도교수 : 유선용

정주희
(인)



전남대학교 소프트웨어중심대학사업단장 귀하

산학협력프로젝트(캡스톤디자인) 결과보고서(요약)

프로젝트명	오토피봇 (Auto Fee Bot : 회비 자동 관리 시스템)		
수행기간	2025. 9. 24. ~ 12. 19.	소요예산	802,936원
소요예산 세부내역	-회의비: 300,000원 -(추가지원금)SW활용비: 502,936원(claude max)		
참여인원	구분	인원수	성명(모두 기재)
	교수	1	유선용
	석박사과정		
	학부생	2	정주현, 김성찬
	기업체	1	(멘토) 임성진
	계	4	
추진배경	<p>현재 저희 동아리에서는 회비 관리 업무를 회원별 입금 내역 확인 → 엑셀 기록 → 개별 안내 메시지 발송의 순서로 수작업으로 수행하고 있습니다. 이러한 수작업 방식은 반복적이고 시간이 많이 소요될 뿐만 아니라, 기록 누락이나 입력 오류 발생 가능성 등 관리상의 여러 문제를 내포하고 있습니다. 저희는 현재 동아리 임원으로 활동하며 직접 회비 관리 업무를 수행하면서, 수작업으로 인한 업무 부담과 안내 누락 위험을 직접 경험하였습니다. 특히 동아리원 수가 증가함에 따라 관리 업무가 점점 복잡해지고, 담당자의 업무 효율성이 현저히 저하되는 문제를 뚜렷하게 체감하였습니다. 더욱이, 시중에서 제공되는 회비 관리 서비스는 엑셀 파일을 통한 미납자 자동 분류 기능 등만 제공하고 싶어, 동아리 운영 현실에 맞는 효율적인 관리 방안을 제공하지 못하고 있습니다.</p> <p>이에 따라, 입금 알림 자동 수집 및 파싱, 회원별 회비 납부 현황 자동 기록, 미납자 대상 자동 안내 메시지 발송 기능을 갖춘 시스템을 구축함으로써, 관리 업무 효율성을 극대화하고 기록 오류 및 안내 누락을 최소화할 필요가 있습니다. 이러한 시스템 도입을 통해, 동아리 운영의 정확성과 효율성을 동시에 향상시킬 수 있을 것으로 기대됩니다.</p>		
목표 및 내용	<p>본 프로젝트의 목표는 동아리 회비 관리 업무를 효율적으로 자동화하는 시스템을 구축하는 것입니다. 기존에는 회원별 입금 내역 확인, 엑셀 기록 작성, 개별 안내 메시지 발송을 모두 수작업으로 수행하였기 때문에 반복적인 업무 부담이 크고, 기록 누락과 오류 발생 가능성이 존재했습니다. 특히, 동아리 임원으로 활동하면서 이러한 문제를 직접 경험한 바 있어, 효율적이고 안정적인 관리 시스템의 필요성을 절감하게 되었습니다. 이에 본 프로젝트에서는 다음과 같은 기능을 구현합니다.</p> <ol style="list-style-type: none"> 1. 입금 알림 자동 수집 및 파싱: 백그라운드 애플리케이션을 구현하여 모바일 입금 내역 알림을 실시간으로 수집하고 파싱합니다. 이를 통해 납부자와 금액을 자동으로 확인하고 회비 납부 현황을 즉시 파악할 수 있습니다. 2. 회원별 회비 납부 현황 자동 기록: 수집된 입금 정보를 자동으로 데이터베이스에 저장하고, 회원별 납부 상태를 실시간으로 관리할 수 있도록 합니다. 3. 미납자 자동 안내 메시지 발송: 납부하지 않은 회원에게 자동으로 안내 메시지를 발송하여, 관리자가 개별적으로 연락해야 하는 부담을 최소화합니다. <p>최종적으로 이러한 기능을 통합하여 회비 관리 자동화 시스템을 구현함으로써, 관리 업무 효율성을 극대화하고 오류 발생 가능성을 최소화하는 것을 목표로 합니다. 또한 프로젝트 수행 과정에서 백엔드 데이터 처리, 자동화 로직 구현, 알림 서비스 연동 등 실무형 개발 역량을 향상시키고, 팀 단위 협업을 통해 문제 해결 능력과 프로젝트 관리 역량을 동시에 강화하는 것을 학습 성과로 설정하였습니다.</p>		
기대효과	<ul style="list-style-type: none"> - 기존 수작업 기반의 반복적 업무 부담 완화 - 회원별 입금 내역 및 회비 관리 기록의 정확성 향상 - 입금 미확인 회원에 대한 자동 알림으로 회비 미납률 감소 - 운영진의 관리 시간 단축과 업무 효율 극대화 		

1. 프로젝트 개요

프로젝트명	오토피봇 (Auto Fee Bot : 회비 자동 관리 시스템)
주제영역	<input type="checkbox"/> 생활 <input type="checkbox"/> 업무 <input type="checkbox"/> 공공/교통 <input type="checkbox"/> 금융/핀테크 <input type="checkbox"/> 의료 <input type="checkbox"/> 교육 <input type="checkbox"/> 유통/쇼핑 <input type="checkbox"/> 엔터테인먼트
기술분야	<input type="checkbox"/> IoT <input type="checkbox"/> 모바일 <input type="checkbox"/> 데스크톱 SW <input type="checkbox"/> 인공지능 <input type="checkbox"/> 보안 <input type="checkbox"/> 가상현실 <input type="checkbox"/> 빅데이터 <input type="checkbox"/> 자동제어기술 <input type="checkbox"/> 블록체인 <input type="checkbox"/> 영상처리 <input type="checkbox"/> 기타()
성과목표	<input type="checkbox"/> 논문게재 및 포스터발표 <input type="checkbox"/> 앱등록 <input type="checkbox"/> 프로그램등록 <input type="checkbox"/> 특허 <input type="checkbox"/> 기술이전 <input type="checkbox"/> 실용화 <input type="checkbox"/> 공모전(<i>공모전명</i>) <input type="checkbox"/> 기타()

2. 프로젝트 추진배경

① 핀테크 기반 모임 통장의 진화와 관리 업무의 사각지대

과거 불투명하고 번거로운 개인 계좌 기반의 수기 관리 방식에서 벗어나기 위해 금융권에서는 다양한 시도가 이어져 왔다. 대표적으로 2018년 12월, 카카오뱅크는 모임장이 아닌 멤버들도 실시간으로 입출금 내역을 확인할 수 있는 '모임통장' 서비스를 출시하며 회비 관리의 투명성 문제를 획기적으로 개선하였다. 이후 모임의 규모가 커짐에 따라 총무 1인에게 집중되는 업무 과중과 금전적 책임 문제를 해결하기 위해, 토스뱅크는 공동 명의자 모두가 카드 발급 및 결제 권한을 갖는 '공동 모임통장'을 통해 권한과 책임의 분산을 제시하기도 하였다.

그러나 이러한 금융 플랫폼의 혁신은 자금의 보관과 내역 공유라는 금융적 기능에 국한되어 있다. 실제 총무가 수행해야 하는 '회비 납부 내역과 회원 명부의 대조', '미납자 식별 및 독촉', '회계 장부 정리'라는 실무적·행정적 절차는 여전히 기술의 사각지대에 놓여 있으며, 이는 고스란히 관리자의 수작업 몫으로 남아있다.

② 수기 관리 프로세스의 구조적 한계와 비효율성

금융 앱이 발전했음에도 불구하고, 대다수의 대학 동아리 및 모임에서의 관리 프로세스는 여전히 '은행 앱 확인 → 엑셀 수기 기록 → 미납자 개별 연락'이라는 아날로그 방식을 답습하고 있다. 이는 다음과 같은 구조적인 문제점을 야기한다.

·반복 업무의 비효율성: 금융 앱의 내역은 단순한 '입금 로그'일 뿐, '누가 어느 달의 회비를 냈는지'를 알려주지 않는다. 총무는 매번 입금 내역을 엑셀 명부와 일일이 대조해야 하며, 이는 막대한 시간과 노력이 소요되는 단순 반복 노동이다.

·인적 오류의 위험: 수작업으로 데이터를 이관하고 입력하는 과정에서 기록 누락이나 오기입이 발생할 가능성이 높다. 아무리 통장이 투명하게 공개되어도 이를 정리하는 장부가 부정확하다면 모임 재정의 신뢰도는 저하될 수밖에 없다.

·커뮤니케이션의 누수와 감정 소모: 미납자를 식별한 후에도 개별적으로 연락을 취하는 과정에서 메시지가 누락되거나, 금전 관계 특성상 발생하는 불필요한 감정 소모가 관리자에게 큰 부담으로 작용한다.

③ 조직 규모 확대에 따른 '관리 복잡성'의 기하급수적 증가

모임 회원의 수가 증가할수록 관리해야 할 데이터의 양과 복잡도는 기하급수적으로 늘어난다. 실제 본 연구진이 학생회 총무부와 동아리 임원진으로서 경험한 바에 따르면, 약 200명 규모

조직의 회비 납부 여부를 확인하고 정리하는 데만 매회 3시간 이상이 소요되었다. 특히 동명이인의 입금, 입금자명 불일치, 회비 금액 상이 등 예외 상황이 발생할 경우, 수작업 관리의 한계는 명확히 드러나며, 이는 관리 담당자의 업무 효율성을 급격히 저하시키는 주요 원인이 된다. 즉, 카카오와 토스가 열어준 '공유의 장' 위에서 실제 운영의 짐은 여전히 사람이 짊어지고 있는 실정이다.

④ 기존 상용 서비스의 부적합성 및 맞춤형 솔루션의 부재

현재 시중에는 회비 관리를 돕는 다양한 서비스나 엑셀 템플릿이 존재하지만, 대부분은 사용자가 직접 내역을 입력해야 하는 '보조 도구' 수준이거나 단순한 미납자 분류 기능에 그치고 있다. 이는 각 모임이 가진 고유한 회칙, 다양한 납부 주기, 미납자 대응 방식 등의 특수성을 반영하지 못해 실제 운영 현실에 맞는 효율적인 대안이 되지 못하고 있다. 따라서 단순한 금융 내역의 공유를 넘어 관리 업무 자체를 자동화할 수 있는 시스템 구축이 시급하다.

⑤ '오토피봇(AutoFeeBot)'을 통한 운영 혁신과 목표

본 연구는 이러한 문제의식을 바탕으로 금융 앱의 알림을 자동으로 수집·파싱하여 수작업 입력을 최소화하는 '자동 회비 관리' 시스템을 제안한다. 이 시스템은 1) 입금 알림의 실시간 자동 수집, 2) 회원 DB와 입금 내역의 자동 매칭 및 기록, 3) 미납자에 대한 조건별 자동 안내 메시지 발송이라는 세 가지 핵심 기능을 통합한다. 이를 통해 기존 수작업 관리에서 필연적으로 발생하던 업무 부담과 인적 오류를 획기적으로 줄이고, 운영의 효율성과 회계 기록의 정확성을 동시에 달성하는 것을 최종 목표로 한다.

3. 프로젝트(주제) 목표 및 내용

본 프로젝트의 목표는 동아리 회비 관리 업무를 효율적으로 자동화하는 시스템을 구축하는 것입니다. 기존에는 회원별 입금 내역 확인, 엑셀 기록 작성, 개별 안내 메시지 발송을 모두 수작업으로 수행하였기 때문에 반복적인 업무 부담이 크고, 기록 누락과 오류 발생 가능성이 존재했습니다. 특히, 동아리 임원진으로 활동하면서 이러한 문제를 직접 경험한 바 있어, 효율적이고 안정적인 관리 시스템의 필요성을 절감하게 되었습니다. 이에 본 프로젝트에서는 다음과 같은 기능을 구현합니다.

1. 입금 알림 자동 수집 및 파싱: 백그라운드 애플리케이션을 구현하여 모바일 입금 내역 알림을 실시간으로 수집하고 파싱합니다. 이를 통해 납부자와 금액을 자동으로 확인하고 회비 납부 현황을 즉시 파악할 수 있습니다.
2. 회원별 회비 납부 현황 자동 기록: 수집된 입금 정보를 자동으로 데이터베이스에 저장하고, 회원별 납부 상태를 실시간으로 관리할 수 있도록 합니다.
3. 미납자 자동 안내 메시지 발송: 납부하지 않은 회원에게 자동으로 안내 메시지를 발송하여, 관리자가 개별적으로 연락해야 하는 부담을 최소화합니다.

최종적으로 이러한 기능을 통합하여 회비 관리 자동화 시스템을 구현함으로써, 관리 업무 효율성을 극대화하고 오류 발생 가능성을 최소화하는 것을 목표로 합니다. 또한 프로젝트

수행 과정에서 백엔드 데이터 처리, 자동화 로직 구현, 알림 서비스 연동 등 실무형 개발 역량을 향상시키고, 팀 단위 협업을 통해 문제 해결 능력과 프로젝트 관리 역량을 동시에 강화하는 것을 학습 성과로 설정하였습니다.

4. 시스템 구성 및 내용



□ 모바일 기반 입금 데이터 수집 및 파싱 기술의 구현

① 안드로이드 NotificationListenerService를 통한 실시간 입금 알림 감지

금융 API 연동의 복잡성을 우회하기 위해 안드로이드가 제공하는 공식 API인 NotificationListenerService를 활용했습니다. 이 서비스는 사용자의 스마트폰에 수신되는 모든 푸시 알림을 백그라운드에서 감시하는 역할을 수행합니다. Kotlin 언어로 개발된 리스너는 onNotificationPosted() 메서드를 오버라이드하여 매 알림 수신 시 해당 알림의 발신처 패키지명(예: com.viva.republica - 토스뱅크)을 확인합니다. 금융 앱으로부터의 알림만을 선별한 후, notification.extras에서 'android.text' 필드를 추출하여 알림 메시지의 원본 텍스트를 획득하는 구조로 설계했습니다.

② 정규표현식 기반 파싱 엔진을 통한 비정형 데이터 구조화

금융 앱의 푸시 알림은 비정형 텍스트 형태로 수신되므로, 이로부터 의미 있는 구조화된 데이터를 추출하는 과정이 필수적입니다. 본 연구는 Java의 Pattern과 Matcher 클래스를 활용한 정규표현식 기반의 동적 파싱 엔진을 개발했습니다. 토스뱅크의 입금 알림 형식인 "[토스] 홍길동님 15,000원 입금"을 예로 들면, 송금자명 추출을 위한 패턴 (?:\w{2,3}[\w+]\w{2,3})?([가-힣\w]+?)님, 거래 금액 추출을 위한 패턴 ([0-9,]+)원, 그리고 거래 시간 추출을 위한 패턴 (\w{2}:\w{2})을 설계하여 적용했습니다. 이러한 패턴들을 조합 적용하면 비정형 텍스트가 JSON 형식의 구조화된 데이터 객체로 변환되며, {senderName: "홍길동", amount: 15000, timestamp: "14:30", source: "토스"} 형태의 정형화된 데이터가 생성됩니다.

③ 모바일 앱과 백엔드 서버 간의 안전한 통신 구현

파싱된 데이터는 Retrofit2 라이브러리를 통해 HTTP POST 요청으로 Spring Boot 백엔드 서버로 전송됩니다. 네트워크의 일시적 오류에 대응하기 위해 Retry Interceptor를 적용하여 전송 실패 시 자동으로 재시도하도록 구성했습니다. 개인 금융 정보 보호를 위해 HTTPS 프로토콜을 강제하였으며, 요청 헤더에 JWT 토큰을 포함시켜 인증된 사용자로부터의 요청만 수락하도록 설계했습니다.

□ 백엔드 서버 아키텍처 및 데이터베이스 구축

① Spring Boot 기반 RESTful API 서버의 설계 및 구현

백엔드 서버는 Spring Boot 3.0을 기반으로 개발되었으며, 모바일 앱으로부터 수집된 입금 데이터를 처리하는 RESTful API를 제공합니다. /api/deposits/parse 엔드포인트에서는 모바일 앱에서 전송된 파싱된 입금 데이터를 수신하고, /api/members/search에서는 데이터베이스의 회원 정보를 조회하며, /api/payments/match에서는 입금 데이터와 회원 정보의 자동 매칭을 수행합니다. 각 엔드포인트는 명확한 요청-응답 구조를 가지며, HTTP 상태 코드(200 성공, 400 잘못된 요청, 401 인증 실패, 500 서버 오류 등)를 활용하여 처리 결과를 클라이언트에 명확하게 전달합니다.

② 회원-입금 자동 매칭 알고리즘의 구현 및 한계 분석

시스템의 핵심 기능은 모바일 앱에서 수집한 입금 데이터를 데이터베이스의 회원 정보와 자동으로 매칭하는 것입니다. 현재 구현된 알고리즘은 '송금자명'을 유일한 식별자로 사용하여 데이터베이스의 Users 테이블에서 해당 회원을 검색합니다. 검색 결과가 정확히 1명일 경우 매칭 성공으로 처리하여 Payments 테이블에 입금 기록을 저장합니다. 그러나 Birthday Paradox 방법론에 따르면 50명 규모의 집단에서 동명이인이 발생할 확률은 약 30~40%에 달하므로, 동명이인이 존재하거나 검색 결과가 0명일 경우 '미확인 입금(Unmatched)' 상태로 분류하여 관리자의 수동 확인을 요청하도록 설계했습니다.

③ MariaDB 데이터베이스 설계 및 구축

데이터의 무결성을 보장하기 위해 MariaDB 10.6을 채택했습니다. 핵심 테이블들은 다음과 같습니다: Users 테이블(user_id, name, phone_number, meeting_id 등 회원 정보), Meetings 테이블(meeting_id, meeting_name, created_date 등 모임 정보), Payments 테이블(payment_id, user_id, amount, payment_date, status 등 입금 내역), NotificationLogs 테이블(log_id, payment_id, message, sent_date 등 발송 이력). 각 테이블 간에는 외래키(Foreign Key) 제약조건을 설정하여 데이터의 참조 무결성을 보장했으며, name, payment_date 등 자주 조회되는 필드에 인덱스를 생성하여 쿼리 성능을 최적화했습니다.

④ Docker 컨테이너 기반 배포 환경 구축

개발 환경과 운영 환경 간의 환경 차이로 인한 오류를 방지하기 위해 Docker 컨테이너 기술을 도입했습니다. Spring Boot 애플리케이션과 MariaDB를 각각의 독립된 컨테이너로 구성하고, docker-compose.yml 파일을 작성하여 두 컨테이너가 같은 Docker 네트워크 상에서 자동으로 통신하도록 설정했습니다. 이러한 구성으로 로컬 개발 환경에서 테스트한 동일한 상태를 AWS EC2 서버에 그대로 배포할 수 있으므로, 배포 오류를 획기적으로 줄이고 배포 속도를 향상시킵니다.

□ 프론트엔드 웹 대시보드 및 보안 시스템 개발

① React 기반 관리자 대시보드 UI/UX 설계

사용자 인터페이스는 React 18 버전으로 개발되었으며, 관리자가 데이터에 기반한 빠른 의사 결정을 내릴 수 있도록 시각적 정보 제공에 중점을 두었습니다. 대시보드의 상단에는 실시간 통계를 카드 형식으로 배치하여 총 회원 수, 납부자 수, 미납자 수, 납부율을 한눈에 파악할 수 있게 했습니다. 중단부에는 Recharts 라이브러리를 활용한 월별 회비 수거 추이 그래프를 시각화하여 시간에 따른 변화 패턴을 직관적으로 이해할 수 있게 구성했습니다. 하단부에는 미납자 목록을 테이블 형식으로 제공하되, 필터링 및 정렬 기능을 추가하여 특정 미납자를 빠르게 검색할 수 있도록 했습니다.

② Apache POI를 활용한 엑셀 호환 기능 구현

기존에 엑셀로 회원 정보를 관리하던 사용자가 새로운 시스템에 거부감 없이 접근할 수 있도록, Apache POI 라이브러리를 백엔드에 통합하여 엑셀 파일 업로드 기능을 개발했습니다. 관리자는 기존에 관리하던 회원 목록 엑셀 파일(예: members.xlsx)을 대시보드에서 업로드하면, 시스템이 각 행의 데이터를 자동으로 파싱하여 데이터베이스에 일괄 등록합니다. 이 과정에서 필수 필드(이름, 전화번호) 검증과 중복 데이터 체크를 수행하여 데이터 품질을 보장합니다.

③ JWT와 OAuth 2.0 기반의 이중 보안 인증 체계 구축

민감한 회비 정보를 다루는 만큼 보안성을 강화했습니다. JWT(JSON Web Token) 기반의 토큰 인증 방식을 채택하여 로그인 후 발급된 토큰을 요청 헤더에 포함시키고, 서버는 서명을 검증하여 토큰의 위변조 여부를 확인합니다. 추가적으로 사용자 편의성을 높이기 위해 Kakao와 Naver의 OAuth 2.0 소셜 로그인 기능을 구현했으므로, 사용자는 별도의 회원가입 절차 없이 기존의 카카오톡이나 네이버 계정으로 로그인할 수 있습니다.

□ 자동화 워크플로우 및 AI 기반 지능형 서비스 개발

① CoolSMS API 연동을 통한 조건부 자동 알림 발송

회비 관리의 실무적 자동화를 완성하기 위해 국내 SMS 서비스 제공자인 CoolSMS의 API를 백엔드에 통합했습니다. 입금 매칭 로직에서 미납을 감지하면, 자동으로 미납자 목록을 필터링하고 해당 인원에 대해 사전에 정의된 템플릿 메시지(예: "안녕하세요. [모임명] [년월] 회비 미납 안내입니다. 입금 부탁드립니다.")를 발송합니다. 이 과정에서 관리자가 일일이 문자를 작성하고 개별 전송하는 반복 업무가 완전히 자동화됩니다. 또한 NotificationLogs 테이블에 발송 기록을 저장하여 추후 감시(audit) 목적으로 활용할 수 있도록 했습니다.

② Gemini API를 활용한 AI 챗봇 구현

단순한 GUI 조작만으로는 사용성 측면에서 한계가 있다고 판단하여, 구글의 Gemini API를 활용한 대화형 AI 챗봇을 시스템에 탑재했습니다. 사용자는 "이번 달 미납자가 몇 명이야?", "1월 회비 수거율 알려줘" 같은 자연어 질의를 통해 데이터베이스에 접근할 수 있습니다. 백엔드에서는 사용자의 질의를 받으면 이를 Gemini API로 전달하되, 동시에 현재 데이터베이스 상태(회원 수, 납부 현황 등)를 컨텍스트로 제공하여 AI가 정확한 답변을 생성할 수 있도록 합니다.

③ 도메인 특화 가드레일(Domain-Specific Guardrails) 프롬프트 엔지니어링 적용

생성형 AI의 환각(Hallucination) 현상, 즉 학습 데이터에 없는 내용을 마치 사실인 것처럼 답

변하는 경향이 금융 데이터를 다루는 시스템에서는 심각한 오류를 초래할 수 있습니다. 이를 방지하기 위해 프롬프트 엔지니어링을 통해 가드레일을 설정했습니다. Gemini API 호출 시 다음과 같은 시스템 프롬프트를 포함시킵니다: "너는 회비 관리 시스템의 AI 어시스턴트이다. 회비 관리, 동아리 운영, 회계 관련 질문에만 답변하고, 다른 주제의 질문에는 '죄송합니다. 이는 회비 관리와 무관한 질문입니다. 회비 현황, 미납자 정보, 수거율 등에 관해 물어봐주세요'라고 답변하라." 이러한 제약을 통해 챗봇이 지정된 도메인 내에서만 활동하도록 강제하며, 신뢰성이 낮은 답변은 사전에 차단합니다.

□ 시스템 통합 및 배포 과정

① 단위 테스트 및 통합 테스트의 수행

각 모듈이 완성된 후에는 JUnit을 활용한 단위 테스트를 수행하여 개별 컴포넌트의 정상 동작을 검증했습니다. 정규표현식 파싱 엔진에 대해서는 다양한 입력 형식의 알림 메시지("[토스] 홍길동님 15,000원 입금", "[카카오뱅크] 김철수 님 5,000원 입금" 등)를 테스트 케이스로 작성하고 각각이 정확히 파싱되는지 검증했습니다. 백엔드 API에 대해서는 Postman을 활용한 HTTP 요청 테스트를 수행했으며, 데이터베이스 쿼리에 대해서는 정상 케이스(존재하는 회원 검색)와 예외 케이스(존재하지 않는 회원 조회, 중복된 입금 기록 등)를 모두 포함시켰습니다. 모든 단위 테스트를 통과한 후, 전체 시스템이 연동되는 통합 테스트를 수행했습니다. 테스트 환경에서 테스트 입금 신호를 생성하여 앱에서 감지 후 서버로 전송, DB에 저장, 대시보드에 반영, SMS 알림 발송에 이르는 전체 파이프라인이 정상 동작하는지 확인했습니다.

② AWS EC2 및 Vercel을 활용한 배포

개발이 완료된 백엔드 애플리케이션은 AWS EC2 인스턴스에 Docker 컨테이너 형태로 배포했습니다. EC2 인스턴스 생성 후 Docker와 docker-compose를 설치한 뒤, 로컬에서 작성한 docker-compose.yml 파일을 EC2에 업로드하여 `docker-compose up -d` 명령으로 서비스를 실행했습니다. 프론트엔드(React 애플리케이션)는 Vercel 플랫폼에 배포했습니다. Vercel은 React 애플리케이션에 최적화된 호스팅 서비스로, GitHub 리포지토리와 연동하여 커밋 시 자동으로 빌드 및 배포되도록 설정했습니다. 이를 통해 개발 과정 중 실시간으로 변경사항을 반영하고 테스트할 수 있는 개발 환경을 구축했습니다.

③ 엔드-투-엔드 자동화 파이프라인의 검증

'사용자 입금(외부) → 모바일 앱이 알림 감지 → 백엔드 서버로 데이터 전송 → DB에 저장 → 웹 대시보드에 반영 → SMS 알림 발송'에 이르는 전체 파이프라인이 사람의 개입 없이 유기적으로 작동하는지 최종 검증했습니다. 각 단계의 로그를 추적하여 데이터가 정상적으로 흐르는지 확인했습니다. 모바일 로그에서 알림 감지 타임스탬프(예: 14:25:33), 백엔드 로그에서 데이터 수신 및 처리 타임스탬프(14:25:34), DB의 Payments 테이블 업데이트 기록(14:25:35), 웹 대시보드의 UI 업데이트(14:25:36), SMS 발송 기록(14:25:37) 등을 모두 확인하여 각 계층이 정상 동작함을 입증했습니다.

5. 프로젝트 결과물에 대한 기술

1) 정성적 연구개발성과(연구개발결과)

본 연구과제인 '오토피봇(Auto Fee Bot)'은 초기 기획 단계에서 설정한 목표에 따라 모바일-서버-웹을 아우르는 통합 회비 관리 자동화 시스템을 성공적으로 구현하였다. 주요 연구개발 성과는 다음과 같다.

① 모바일 기반 입금 알림 데이터 파싱 및 전송 시스템 구축

·알림 리스너 구현: 안드로이드의 NotificationListenerService를 활용하여 토스 등 금융 앱의 푸시 알림을 백그라운드에서 실시간으로 감지하는 기능을 구현하였다.

·정규표현식 파싱 엔진: 비정형 텍스트로 수신되는 알림 메시지에서 '입금자명', '거래 금액', '통장 이름' 등의 핵심 데이터를 추출하는 파싱 로직을 완성하였으며, 이를 통해 별도의 금융 API 없이도 실시간 입금 데이터를 확보하는 기술을 실증하였다.

② 안정적인 백엔드 및 데이터베이스 인프라 구현

·Spring Boot 기반 API 서버: 파싱된 데이터를 수신하고 회원 정보와 매칭하는 RESTful API 서버를 구축하였다.

·컨테이너 기반 배포: Docker를 도입하여 Spring Boot 애플리케이션과 MariaDB를 컨테이너화 함으로써 개발 및 운영 환경의 일관성을 확보하고, 배포 효율성을 극대화하였다.

·데이터 무결성 확보: 회원 정보, 시계열 입금 내역, 알림 발송 로그 등을 관계형 데이터베이스 (MariaDB)에 체계적으로 저장하여 데이터의 영속성과 무결성을 보장하였다.

③ 사용자 친화적 관리 대시보드 및 편의 기능 개발

·React 기반 웹 대시보드: 관리자가 전체 납부 현황, 미납자 목록 등을 직관적으로 확인할 수 있는 반응형 웹 인터페이스를 구현하였다.

·Apache POI 엑셀 연동: 기존 엑셀 관리 방식에 익숙한 사용자를 위해 엑셀 파일을 업로드하여 모임원을 일괄 등록하는 기능을 구현하여 시스템 도입 장벽을 낮추었다.

·CoolSMS 자동 알림 연동: 미납자 발생 시 시스템이 자동으로 SMS를 발송하는 로직을 연동하여 관리자의 반복적인 연락 업무를 자동화하였다.

④ 생성형 AI 기반 지능형 챗봇 도입

·Gemini API 연동: 구글의 Gemini API를 활용하여 자연어 질의가 가능한 챗봇을 탑재하였다.

·도메인 특화 가드레일 적용: 프롬프트 엔지니어링을 통해 회비 관리와 관련된 질문에만 응답하도록 제한하여 정보의 신뢰성을 확보하고 환각 현상을 방지하였다.

2) 목표 달성 수준

전반적인 시스템 구축과 입금 데이터 파싱이라는 핵심 기반 목표는 성공적으로 달성하였으나 완전 자동화된 식별 프로세스와 연속 대화가 가능한 고성능 챗봇 구현이라는 두 가지 핵심 목표는 구조적 및 기술적 한계로 인해 목표 수준에 미치지 못하였다.

① 핵심 아키텍처 구축 및 관리자 인터페이스 완성 목표 (달성도 100%)

연구 초기 목표였던 데이터 수집과 관리자 인터페이스 구축은 완벽하게 구현되었다.

·데이터 파싱 및 수집: Kotlin 기반의 안드로이드 알림 리스너와 정규표현식 파싱 엔진을 성공적으로 구현하여 토스 금융 앱의 푸시 알림을 실시간으로 감지하고, 정형화된 데이터로 변환하

는 목표를 100% 달성하였다.

·시스템 인프라 구축: Spring Boot 기반의 백엔드 API 서버와 MariaDB를 Docker 컨테이너 환경으로 구축하여 개발 환경과 운영 환경의 일관성을 확보하고, 배포 안정성을 높였다. 데이터 저장 및 처리 로직은 모두 정상 동작하여 시스템 구축 목표를 100% 달성하였다.

·관리 기능 및 호환성: React 기반의 관리자 웹 대시보드와 사용자 인증 시스템(JWT, OAuth 2.0)을 완성하였다. 또한, Apache POI 라이브러리를 활용한 엑셀 업로드 기능을 구현하여 기존 엑셀 관리 체계와의 호환성 확보 목표를 100% 달성하였다.

② 자동화 로직 및 지능형 서비스 목표

관리 업무의 비효율성을 해소하고자 했던 자동화 및 지능화 목표는 시스템의 핵심 한계로 인해 부분적인 달성에 그쳤다.

㉠ 회원 매칭 및 알림 자동화 (달성도 약 80%)

회비 관리의 핵심인 '입금 확인 및 알림' 프로세스는 기초 로직 구현은 완료되었으나 완전 자동화에는 미달되었다.

·기본 로직 구현: 입금 알림 데이터와 회원 DB를 대조하여 납부 상태를 업데이트하는 매칭 로직과 CoolSMS를 연동한 미납자 자동 알림 발송 기능은 구현되었다.

·달성 한계: 그러나 입금자명을 유일한 식별자로 사용하는 현재의 구조적 한계로 인해 동명이인이 발생했을 경우, 시스템이 입금 주체를 정확히 식별하지 못하여 자동화에 실패한다. 이는 관리자의 수동 확인 및 수정을 필연적으로 요구하며, 완전 자동화라는 목표 대비 약 80% 수준에 머무르는 결과이다.

㉡ AI 챗봇 성능 및 최적화 (달성도 약 90%)

구글 Gemini API를 활용한 지능형 서비스 구현 목표는 기술 아키텍처의 미완성으로 인해 성능 목표를 충족시키지 못했다.

·기능적 한계 (맥락 소실): 챗봇이 사용자 질의에 대해 단발성 응답은 가능하나 Redis 도입이 미완성됨에 따라 대화 맥락을 유지하지 못하는 한계를 보였다. 이로 인해 연속 질문이나 복합적인 질의 처리 능력이 저하되었다.

·성능적 한계 (응답 지연): 챗봇의 모든 대화 로그 및 데이터 조회를 메모리 기반이 아닌 디스크 기반의 MariaDB에 직접 의존하도록 구현하였다. 이로 인해 데이터 입출력 과정에서 병목 현상이 발생하여 응답 지연이 발견되었으며, 이는 지능형 서비스의 실시간 사용자 경험 목표를 저해하는 원인이 되었다.

구분	기능정의	세부기능 설명
앱	회비 납부 알림 수집 및 확인	- 모바일 입금 알림을 실시간으로 수집 - 입금자 이름, 금액, 날짜를 자동 파싱 - 파싱 데이터를 DB로 전송
백엔드	데이터 처리 및 자동화 관리	- 입금 내역을 데이터베이스에 자동 저장 - 회원별 납부 상태 실시간 관리 - 미납자 목록 자동 생성 및 알림 전송 트리거
웹	관리자용 회비 관리 대시보드	- 회원별 회비 납부 현황 조회

		- 미납자 목록 확인 가능 - 통계/월별 납부 현황 그래프 제공
기타	알림 서비스 연동	- 미납 회원에게 자동 안내 메시지 발송 - SMS/메신저 연동 가능 - 안내 발송 이력 관리

6. 프로젝트 진행내용

1) 참여인원 및 담당 역할

연번	소속학과	성명	수행역할 분담내용
1	컴퓨터정보통신공학과	정주현	요구사항 분석 및 시스템 설계 후 UI 개발
2	컴퓨터정보통신공학과	김성찬	시스템 개발 - 입금 내역 관리 및 기록
3			
4			

2) 회의 및 SW멘토링 진행

번호	일시/장소	회의/멘토링 내용(상세히 작성)	관련 사진
1	2025. 10. 31. (14:00~16:00) / 공대7호관 누리라운지	- 개발 대상 정확히 세분화하여 목표 다시 잡기 - 경쟁사 조사 철저히 하여 사업화에 대한 명분 설명 보완	멘토님과의 사진은 멘토님이 보유하고 계십니다.

7. 프로젝트 세부일정 및 내용

No.	작업 내용	9월				10월				11월				12월				담당자	비고
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4		
1	계획서 작성																	정주현/ 김성찬	
2	요구사항 분석 및 시스템 설계																	정주현/ 김성찬	
3	로그인 페이지 UI 설계																	김성찬	

·활용방안: 본 연구의 핵심 기술인 '알림 리스너 기반 파싱 엔진'을 응용하여 모임의 소비 패턴을 분석하는 가계부 어플리케이션으로 활용 범위를 확장할 수 있다.

3) 추가연구의 필요성 및 타 연구에의 응용

① 금융 특화 LLM 및 가드레일 기술의 고도화

·필요성: 현재 Gemini API와 가드레일을 적용하여 기본적인 회비 질의응답을 구현하였으나, 향후 축적된 회비 데이터를 바탕으로 "다음 달 예산 부족 예측", "불필요한 지출 항목 분석" 등 재무 컨설팅 수준의 AI 에이전트로 발전시키기 위한 추가 연구가 필요하다.

·응용: 본 연구에서 검증된 '도메인 특화 가드레일' 기술은 금융이나 법률 등 데이터의 신뢰성과 보안이 중요한 타 분야의 생성형 AI 도입 연구에 중요한 레퍼런스로 활용될 수 있다.

② 범용 알림 파싱 엔진의 확장

·필요성: 현재는 주요 시중은행(토스)의 알림에 최적화되어 있으나, 증권사, 저축은행, 간편결제(네이버페이 등) 등 다양한 금융 플랫폼의 알림 형식을 학습하고 동적으로 대응하는 '범용 파싱 엔진'으로의 고도화가 요구된다.

·응용: 이 기술은 물류 배송 알림, 예약 확정 알림 등 스마트폰 푸시 알림을 데이터화하여 2차 가치를 창출하는 '모바일 로그 마이닝(Mobile Log Mining)' 연구 분야로 확장 가능하다.

③ 영수증 OCR 및 지출 증빙 자동화

·필요성: 현재의 '수입(입금)' 관리 중심에서 나아가, 영수증 사진을 찍으면 자동으로 금액과 사용처를 인식하여 지출 내역을 등록하는 OCR(광학 문자 인식) 기술 통합 연구가 필요하다. 이를 통해 수입과 지출을 아우르는 완전한 회계 자동화 시스템을 완성한다.

4) 기업화 및 기술이전 추진방안

① SaaS(Software as a Service) 형태의 사업화 모델

·Freemium 전략: 소규모 동아리(회원 30명 이하)에는 무료로 기본 기능을 제공하여 초기 사용자를 확보하고, 대규모 인원 관리와 SMS 대량 발송, AI 심화 분석 기능이 필요한 학생회나 연합 동아리에는 월 구독료를 부과하는 수익 모델을 수립한다.

·B2B 제휴: 대학 본부의 학생지원팀이나 총학생회와 제휴를 맺어 교내 공식 동아리 관리 표준 플랫폼으로 납품하고 기술 지원 비용을 받는 B2B(Business to Business) 모델을 추진한다.

② 오픈소스 공개 및 생태계 조성

·본 프로젝트의 핵심인 '은행 알림 파싱 정규식 라이브러리'를 오픈소스로 공개한다. 이를 통해 외부 개발자들의 기여를 유도하여 다양한 은행의 알림 패턴을 빠르게 확보하고, 시스템의 안정성을 검증받아 기술적 신뢰도를 확보한다.

③ 지식재산권 확보 및 기술이전

'모바일 알림 리스너를 활용한 비연동형 금융 거래 내역 집계 시스템'에 대한 특허 출원을 검토하여 독자적인 기술 권리를 확보한다. 향후 핀테크 스타트업이나 가계부 앱 개발사 등에 해당 파싱 엔진 및 AI 가드레일 모듈을 기술 이전하거나 M&A의 대상으로 가치를 인정받을 수 있도록 한다.

9. 참고자료 및 문헌

1) Automated Extraction of Personal Knowledge from Smartphone Push Notifications

저자: Yuanchun Li, Ziyue Yang, Yao Guo, Xiangqun Chen, Yuvraj Agarwal, Jason I. Hong
(IEEE International Conference on Big Data, Big Data 2018)

2) Design and Implementation of Financial SMS Extraction and Analysis

저자: Shlok Gujar, Yash Kokate, Govind Dalwani

International Journal of Advances in Electronics and Computer Science, Volume 5 Issue 11)