[QR 피싱 여부 판별 서비스]

요구사항 정의서

2025년 10월 17일

문서번호 : 2025-13-01
소　　속 : 충북대학교 소프트웨어학부
팀　　명 : 404 FOUND
팀　　원 : (팀장)김여민(2024042053),
심연우(2024042050),
최수연(2024042016)

제/개정 이력

버전	날짜	작성자 성명	제/개정사항	비 고

목 차

1. 서 론

1.1 문서 목적 및 범위

본 문서는 QREX(QR eXamination) 웹 서비스의 개발에 필요한 모든 소프트웨어 요구사항을 정의합니다. 이는 서비스의 기능적 요구사항 및 비기능적 요구사항을 명확히 하고, 시스템의 범위와 경계를 설정하는 것을 목적으로 합니다.

1.2 프로젝트 개요

1.2.1 프로젝트 정의

본 QREX 프로젝트는 최근 증가하고 있는 QR 코드 기반의 피싱 범죄로부터 사용자를 보호하기 위해 고안된 웹 서비스입니다.

2.2.2 주요 기능 설명

QREX는 피싱 여부가 의심되는 QR을 스캔하여, QR 속 사이트가 유해한 사이트인지를 판별하는 웹 서비스입니다. 사용자는 자신이 검사한 결과를 다시 확인할 수 있으며, 게시판을 통해 피싱 사례를 공유하고 의견을 나눌 수 있습니다.

1. RAG 기반의 QR/URL 실시간 분석 : 카메라나 갤러리 이미지를 통해 QR 코드를 인식하고, URL의 피싱 여부를 즉시 판별.
2. 개인 기록 관리 : 사용자가 검사한 QR 코드의 기록을 저장하고 언제든지 다시 확인 가능.
3. 커뮤니티 : 사용자들이 게시판을 통해 피싱 사례 공유 가능.

1.3 용어 정의

본 문서의 이해를 돕기 위해 사용된 모든 용어 및 약어를 설명하고 정의합니다.

용어	설명
QR 피싱	Quick Response Phising. 악성 코드가 포함된 QR 코드 또는 사용자를 피싱 사이트로 유도하는 URL이 삽입된 QR 코드를 사용하는 사기 행위
RAG	Retrieval-Augmented Generation(검색 증강 생성)의 약어. 최신 피싱 사례데이터 등 외부 지식을 검색하여 URL 판별의 정확도를 높이는데 사용되는 기술 개

	념
FastAPI	백엔드 서버 구축에 사용되는 Pyrhon 기반 웹 프레임 워크. RESTful API 통신을 담당
RESTful API	프론트엔드와 백엔드가 데이터를 주고받는 통신 방식. HTTP 메서드와 URL을 사용하는 표준 규격
토큰(Token)	사용자가 로그인 후 서버로부터 발급받는 인증 정보로, 서버에 페이지 접근 권한을 증명할 때 사용한다.
애자일(Agile)	소프트웨어 개발 방법론의 한 종류로, 계획보다는 실행과 변화에 민첩하게 반응하는 유연하고 반복적인 개발 방법
스크럼(Scrum)	애자일 방법론의 구체적인 실행 프레임워크. 짧은 주기의 반복 개발과 팀 중심의 협업을 강조

1.4 참조 문서

1. 시스템 정의서 - 프로젝트 기능 및 계획 정의 문서
2. Github README - QREX 프로젝트 정의 및 기능 정리 문서
3. 시스템 아키텍처 - 사용자, 프론트, 백, 분석 서버, db 간의 상호작용 다이어그램
4. 기능적 요구사항 - 백엔드와 프론트엔드의 각 기능별 상세 요구사항
5. 데이터베이스 스키마 - USER, ANALYSIS, BOARD, COMMENT 테이블의 관계 및 컬럼 정보
6. 회의록 - 팀 미팅 내용, 주요 결정 사항 및 진행 상황 기록
7. Figma UI 디자인 - 사용자 인터페이스의 전체적인 레이아웃 및 디자인

2. 기능적 요구사항

2.1 기능적 요구사항

F1. QR코드 및 URL 분석 기능

FR-101 사용자가 QR 코드 이미지를 업로드하여 분석을 요청한다.

FR-102 시스템은 분석 요청된 URL에 대한 피싱 위험 여부를 제공해야 한다.

FR-103 시스템은 판별 결과와 함께 해당 URL의 기본 정보를 사용자에게 제공해야 한다.

FR-104 시스템은 판별 결과를 안전, 주의, 위험 등급 중 하나로 명확히 표시해야 한다.

F2. 사용자 계정 및 개인화 기능

FR-201 사용자는 서비스 이용을 위해 회원가입을 할 수 있어야 한다.

FR-202 사용자는 등록된 개인 정보를 수정할 수 있어야 한다.
FR-203 사용자는 과거에 검사한 모든 QR 기록을 조회할 수 있어야 한다.
FR-204 사용자는 조회된 검사 기록 목록에서 원하는 기록을 선택하여 삭제할 수 있어야 한다.

F3. 커뮤니티 및 소통 기능
FR-301 사용자는 피싱 사례를 공유하는 게시판을 이용할 수 있어야 한다.
FR-302 사용자는 게시판에 새로운 게시글을 작성하고 등록할 수 있어야 한다.
FR-303 사용자는 게시글에 대한 댓글을 작성할 수 있어야 한다.
FR-304 사용자는 부적절한 게시글이나 댓글을 신고할 수 있어야 한다.
FR-305 사용자는 본인이 작성한 게시글과 댓글 중 원하는 기록을 선택하여 삭제할 수 있어야 한다.

2.2 비기능적 요구사항

1) 운영 환경에 대한 요구사항
- NF-101 시스템은 표준 웹 브라우저 환경에서 OS와 무관하게 동작할 수 있어야 한다.
- NF-102 시스템은 Chrome, Safari, Edge 등 최신 주요 웹 브라우저 환경에서 정상적으로 동작해야 한다.
- NF-103 QR 코드 분석 기능은 JPEG, PNG 형식의 이미지 파일을 처리할 수 있어야 한다.
- NF-104 메인 비즈니스 로직과 AI 기반 분석 로직은 모두 Spring Boot 환경에서 수행되어야 한다.
- NF-105 GeoIP 조회를 위해 MaxMind GeoLite2 데이터베이스를 로컬 환경에 구성해야 한다.

2) 성능 요구사항
- NF-201 QR/URL 분석 요청 후 피싱 판별 결과를 화면에 표시하는 응답시간은 최대 5초를 넘지 않아야 한다.
- NF-202 사용자 검사 기록 조회 시, 목록 로딩 시간을 최대 5초를 넘지 않아야 한다.
- NF-203 페이지 조회 및 이동 시, 화면 전환 시간은 최대 2초를 넘지 않아야 한다.
- NF-204 IP 위치(GeoIP) 조회는 외부 API 호출 없이 로컬 데이터베이스를 통해 실시간으로 이루어져야 한다.

3) 보안 요구사항
- NF-301 사용자 회원가입 시 비밀번호는 암호화되어 저장되어야 한다.
- NF-302 사용자가 로그인 후 발급받은 토큰의 유무에 따라 페이지 접근 가능 정도가 달라야 한다.
- NF-303 URL 분석 과정에서 악성 스크립트 실행을 방지하도록 안전하게 처리되어야 한다.

4) 문화 및 정책적 요구사항

- NF-401 게시판 이용에 대한 운영 정책을 마련하고 사용자에게 공지해야 한다.
- NF-402 피싱 판별 알고리즘에 사용되는 데이터 출처 및 분석 기준에 대한 투명성을 확보해야 한다.

2.3 인터페이스 요구사항

1) 사용자 인터페이스 요구사항
- IR-101 QREX는 모바일 및 데스크톱 환경에 모두 적합한 반응형 웹 UI를 제공해야 한다.
- IR-102 피싱 판별 결과는 색상 기반의 시각적 지표를 사용하여 명확히 구분되어야 한다.
- IR-103 모든 화면의 요소는 쉽게 조작할 수 있도록 충분한 크기와 간격을 확보해야 한다.
- IR-104 서비스 전체적으로 일관된 디자인 테마 및 레이아웃을 적용하여 사용성을 높인다.

2) 외부 시스템 인터페이스 요구사항
- IR-201 URL 분석과 AI 기반 판별 기능은 Spring Boot 서버 내에서 수행되며, 필요한 경우 외부 API와 RESTful 방식으로 통신해야 한다.
- IR-202 사용자는 Kakao 로그인 및 Google 로그인 등 외부 소셜 인증 서비스와의 연동을 통해 회원 인증을 수행할 수 있어야 한다.
- IR-203 텍스트 기반 데이터(게시글, 댓글, 검사 기록 등)는 MySQL 데이터베이스에 저장되어야 하며, 이미지 파일 등 대용량 파일은 클라우드 스토리지(예: Firebase Storage, AWS S3 등)에 저장할 수 있어야 한다.
- IR-204 Spring Boot 서버는 IP 위치(GeoIP)조회를 위해 MaxMind GeoLite2 로컬 DB에 직접 접근해야 한다.
- IR-205 Spring Boot 서버는 Safe Browsing 결과를 획득하기 위해 Google Safe Browsing API를 직접 호출해야 한다.

3. 기타 요구사항

- OR-101 프로젝트의 모든 산출물은 GIt 기반 버전 관리 시스템을 사용하여 관리되어야 한다.
- OR-102 서비스 개발 팀은 협업 및 일정 관리를 위해 스크럼(Scrum) 등 애자일(Agile) 방법론을 채택하고 준수해야 한다.
- OR-103 RAG 기반 AI 분석은 Spring Boot 서버 내부에서 전처리된 데이터를 기반으로 수행되어야 하며, 분석 과정에서 필요한 외부 API는 Spring Boot에서 직접 호출해야 한다.
- OR-104 모든 최종 분석 결과는 Spring Boot 서버에서 분석 프로세스를 완료한 후

QR_Analysis_History 테이블에 즉시 저장되어야 한다.

4. 참고문헌 및 부록

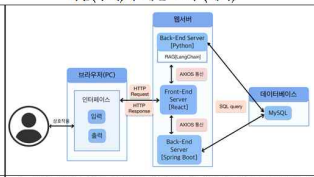
1. 시스템 정의서

시스템 정의서(프로젝트 요약서)

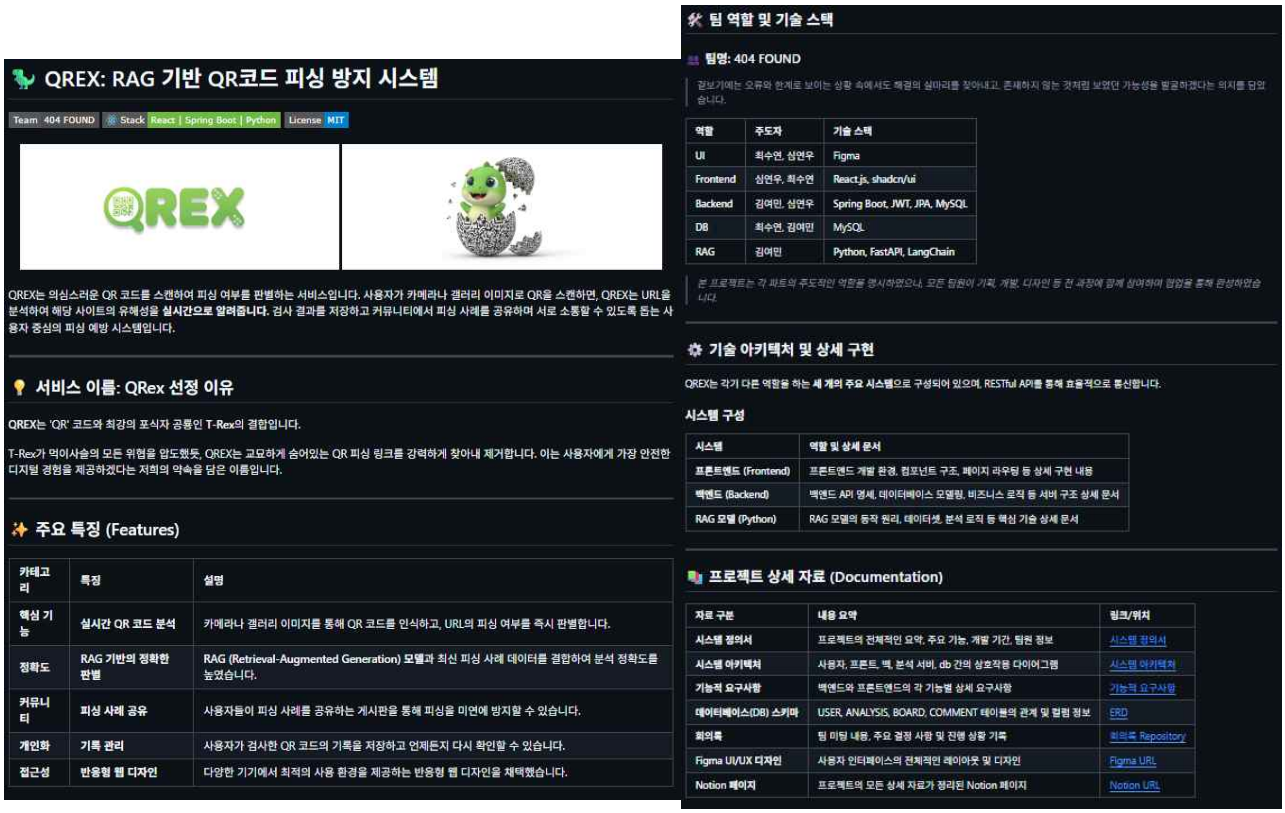
(404 FOUND) 팀

작품명 (주제)	(국문) QR 피싱 여부 판별 웹 사이트		
	(영문) QR Phishing Checker Website		
책 임 자 (팀장)	성 명	김여민	
	소 속	소프트웨어학부	
	학 번	2024042053	
개발기간	2025 년 9 월 5 일 ~ 2025 년 12 월 12 일		
참여학생	학번	이름	전공
	2024042016	최수연	소프트웨어학부
	2024042050	심연우	소프트웨어학부
지도교수	강재구		
작품(주제)에 대한 요약			
작품 설명	QREX는 피싱 여부가 의심되는 QR을 스캔하여 QR 속 사이트가 유해한 사이트인지를 판별한다. 자신이 검사한 결과를 다시 확인할 수 있고 게시판을 통해 피싱 사례를 공유하며 피싱에 대해 의견을 공유할 수 있다.		
작품의 주요 기능	1. 카메라 또는 갤러리의 QR 이미지를 인식한다. 2. 인식한 URL 정보를 서버로 전달한다. 3. 서버에서 URL을 분석하여 피싱 여부를 판별한다. 4. 판별 결과와 URL의 기본 정보를 사용자에게 제공한다. 5. 사용자가 검사한 QR 코드 결과를 저장하고 다시 확인할 수 있도록 한다. 6. 게시판을 통해 사용자들이 피싱 사례를 공유할 수 있도록 한다. 7. 회원가입 시 입력한 개인정보를 수정할 수 있도록 한다. 8. 댓글 신고 기능을 제공하여 건전한 게시판 문화를 형성한다.		

작품(주제)에 대한 요약 (계속)

운영개념	
기타 개발 시 고려사항	1. 피싱 여부 판별 결과를 신뢰할 수 있도록 RAG 기반의 지식 검색 및 최신 피싱 사례 데이터를 결합하여 분석 정확도를 높인다. 2. QR 스캔 URL과 사용자 정보를 안전하게 관리한다.
오픈소스활용 및 기여 방안	1. UI/프론트엔드 라이브러리를 가져와 디자인 적용 2. 양성 및 악성 QR 코드 데이터셋 활용
선행기술 조사 분석	<특허> 1. KR101115250B1(김광태, 안영택, 강유진) / 1020110080168 : QR코드의 안전도 검사 장치 및 방법 2. 1020170097491 (2017.08.01.) 주식회사 에프원시큐리티 머신러닝을 이용한 악성코드 유포지 위험도 분석 시스템 <논문> 1. 한국정보통신학회논문지 제26권 제 12호, 2022년, pp. 1,786 - 1,793, 김영준, 이재우 URL 주요특징을 고려한 악성URL 머신러닝 탐지모델 개발 2. 한국인터넷방송통신학회 논문지 v.12 no.1, 2012년, pp.83 - 89, 양형규, QR 코드의 보안 취약점과 대응 방안 연구 3. 이동건, 황규준, 김동오, 황진석, 「큐싱(Qshing) 해킹 대응 방안 연구」, 『한국IT정책경영학회 논문지』, 한국IT정책경영학회, 2024, 37 - 42 p. 4. 신현창, 이주형, 김종민, 「큐싱(Qshing) 공격 탐지를 위한 시스템 구현」, 『융합보안논문지』, 한국융합보안학회, 2023, 55 - 61 p. <상용 제품> 1. askurl: https://askurl.io/ 2. KT '안심 QR 서비스' 3. 짝다잡아 :https://play.google.com/store/apps/details?id=com.ps.wb&hl=ko 4. 블랙 쏘세지 :https://play.google.com/store/apps/details?id=io.xrium.blacksawsage&hl=ko
Key Words (5개) : QR, 피싱 예방, 보안 서비스, RAG, 실시간 판별	
지도 교수	강재구 (서명)

2. Github README



QREX: RAG 기반 QR코드 피싱 방지 시스템

Team 404 FOUND | Stack React | Spring Boot | Python | License MIT

QREX는 의심스러운 QR 코드를 스캔하여 피싱 여부를 판별하는 서비스입니다. 사용자가 카메라나 갤러리 이미지로 QR을 스캔하면, QREX는 URL을 분석하여 해당 사이트의 유해성을 **실시간으로 알려줍니다.** 검사 결과를 저장하고 커뮤니티에서 피싱 사례를 공유하며 서로 소통할 수 있도록 돕는 사용자 중심의 피싱 예방 시스템입니다.

서비스 이름: QRex 선정 이유

QREX는 'QR' 코드와 최강의 포식자 공룡인 **T-Rex**의 결합입니다.

T-Rex가 먹이사슬의 모든 위협을 압도했듯, QREX는 교묘하게 숨어있는 QR 피싱 링크를 강력하게 찾아내 제거합니다. 이는 사용자에게 가장 안전한 디지털 경험을 제공하겠다는 저희의 약속을 담은 이름입니다.

주요 특징 (Features)

카테고리	특징	설명
핵심 기능	실시간 QR 코드 분석	카메라나 갤러리 이미지를 통해 QR 코드를 인식하고, URL의 피싱 여부를 즉시 판별합니다.
정확도	RAG 기반의 정확한 판별	RAG (Retrieval-Augmented Generation) 모델과 최신 피싱 사례 데이터를 결합하여 분석 정확도를 높였습니다.
커뮤니티	피싱 사례 공유	사용자들이 피싱 사례를 공유하는 게시판을 통해 피싱을 미연에 방지할 수 있습니다.
개인화	기록 관리	사용자가 검사한 QR 코드의 기록을 저장하고 언제든지 다시 확인할 수 있습니다.
접근성	반응형 웹 디자인	다양한 기기에서 최적의 사용 환경을 제공하는 반응형 웹 디자인을 채택했습니다.

팀 역할 및 기술 스택

팀명: 404 FOUND

검보기에는 오류와 한계로 보이는 상황 속에서도 해결의 실마리를 찾아내고, 존재하지 않는 것처럼 보였던 가능성을 발굴하겠다는 의지를 담았습니다.

역할	주도자	기술 스택
UI	최수연, 심연우	Figma
Frontend	심연우, 최수연	React.js, shadcn/ui
Backend	김여민, 심연우	Spring Boot, JWT, JPA, MySQL
DB	최수연, 김여민	MySQL
RAG	김여민	Python, FastAPI, LangChain

본 프로젝트는 각 파트의 주도적인 역할을 명시하였으나, 모든 팀원이 기획, 개발, 디자인 등 전 과정에 걸쳐 상호하며 협업을 통해 완성하였습니다.

기술 아키텍처 및 상세 구현

QREX는 각기 다른 역할을 하는 **세 개의 주요 시스템**으로 구성되어 있으며, RESTful API를 통해 효율적으로 통신합니다.

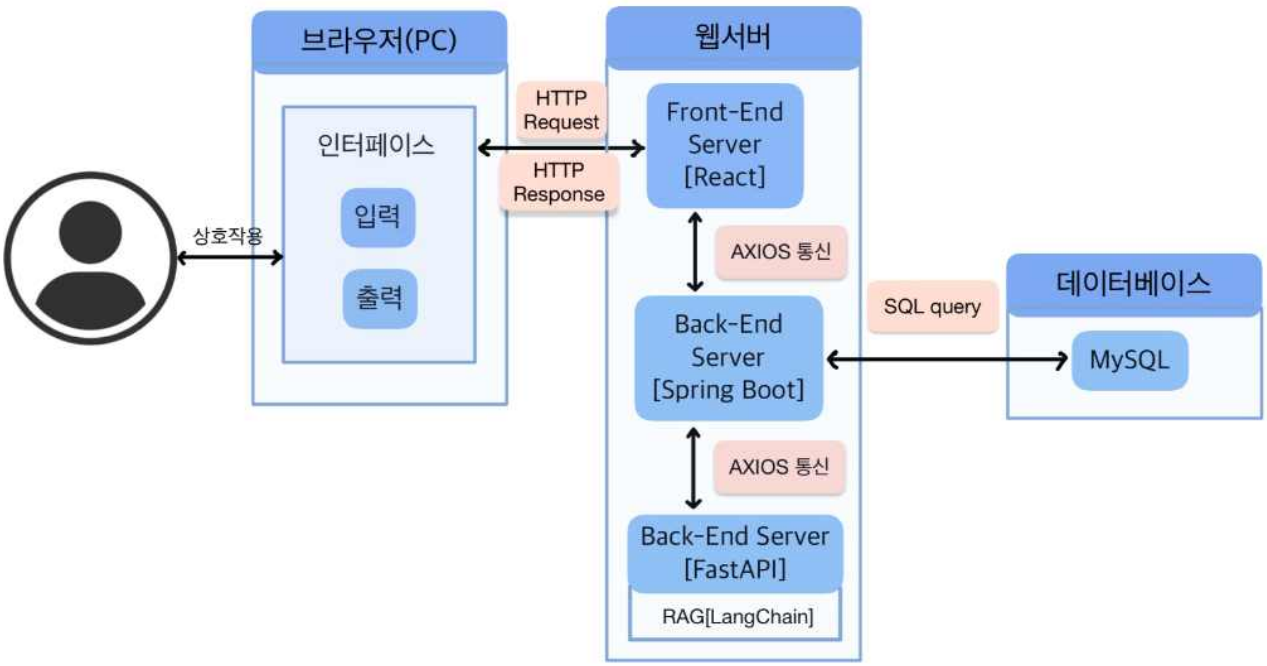
시스템 구성

시스템	역할 및 상세 문서
프론트엔드 (Frontend)	프론트엔드 개발 환경, 컴포넌트 구조, 페이지 라우팅 등 상세 구현 내용
백엔드 (Backend)	백엔드 API 명세, 데이터베이스 모델링, 비즈니스 로직 등 서버 구조 상세 문서
RAG 모델 (Python)	RAG 모델의 동작 원리, 데이터셋, 분석 로직 등 핵심 기술 상세 문서

프로젝트 상세 자료 (Documentation)

자료 구분	내용 요약	링크/위치
시스템 정의서	프로젝트의 전체적인 요약, 주요 기능, 개발 기간, 팀원 정보	시스템 정의서
시스템 아키텍처	사용자, 프론트, 백, 분석 서버, db 간의 상호작용 다이어그램	시스템 아키텍처
기능적 요구사항	백엔드와 프론트엔드의 각 기능별 상세 요구사항	기능적 요구사항
데이터베이스(DB) 스키마	USER, ANALYSIS, BOARD, COMMENT 테이블의 관계 및 컬럼 정보	ERD
회의록	팀 미팅 내용, 주요 결정 사항 및 진행 상황 기록	회의록 Repository
Figma UI/UX 디자인	사용자 인터페이스의 전체적인 레이아웃 및 디자인	Figma URL
Notion 페이지	프로젝트의 모든 상세 자료가 정리된 Notion 페이지	Notion URL

3. 시스템 아키텍처



4. 기능적 요구사항

Front

▼ 헤더

로그인 여부 판단 ← 토큰

- 로고: 누르면 메인페이지로
- analysis: qr scan 페이지로 이동
- my post: my post 페이지로 이동
- community: community 페이지로 이동
- login 버튼: login 페이지로 이동
- 로그인 하지 않고 탭 클릭 → 로그인 요청 팝업 → 로그인 버튼 → 로그인 페이지로 이동

<로그인 상태 시>

- 아바타(프로필) : 탭 창 나타남
  - Tab - Edit profile : 회원 정보 수정 탭
    - Save changed : 수정 내용 저장
  - Tab - Delete account : 회원탈퇴 탭
    - Delete account : 회원 탈퇴 완료
- logout 버튼: 로그아웃 & 메인화면 이동

▼ 메인 페이지

- 마스코트 (공룡) 중앙 배치

▼ 로그인 페이지

- ID 입력
- Password 입력

Front 1

- Login : 로그인 실행
- SignUp : 회원가입 페이지로 이동
- Login in with Google : Google ID 로그인 창으로 이동
- Login in with Kakao : Kakao ID 로그인 창으로 이동

▼ 회원가입 페이지

- name: 이름 입력
- id: Id 입력
- check 버튼: id 중복 확인
- password: password 입력
- cofirm password: password에서 입력한 암호 확인
- sign up 버튼
  - 회원가입 성공 → 성공 알림 팝업→ home 버튼 → 메인 화면으로 이동
  - 회원가입 실패 → 실패 알림 팝업 → try again 버튼 → 다시 sign Up 페이지로 이동

▼ analysis 페이지(qr scan전 → analysis qr → qr scan 후)

- 로그인 하지 않고 탭 클릭 → 로그인 요청 팝업 → 로그인 버튼 → 로그인 페이지로 이동
- qr scan전 페이지(반으로 나눔. 왼쪽에는 qr scan 버튼탭, 오른쪽에는 qr analysis history 탭)
- analyzing qr 페이지: analyzing 값 나오기 전 progress 막대(진행 상태에 따라 막대 바 채워지는 애니메이션 형태로 시각화) 띄워 피싱 여부 판별 진행 중 임을 표시 → 종료 시 qr scan 후 페이지 이동
- qr scan 후(반으로 나눔. 왼쪽에는 qr analysis result 탭, 오른쪽에는 qr analysis history 탭)
- qr scan 아이콘 탭
  - qr 스캔 아이콘: 사진 보관함/사진 찍기/ 파일 선택 탭 중 하나 선택 하여 qr 스캔 → analyzing qr 페이지 이동
    - 사진 보관함 : 갤러리 실행

Front 2

        - 사진 찍기 : 카메라 실행
        - 파일 선택 : 파일 탐색기 실행
    - qr analysis history: 그동안 피싱여부 판별해 보았던 기록 확인
        - qr analysis history 글자: qr analysis history 맨 앞 페이지로 이동
        - title 클릭: 왼쪽에 Title에 해당하는 qr analysis results 페이지 띄움
        - 페이지네이션
    - qr anaysis result
        - 상태: 안전/주의/위험
        - url: qr을 스캔하였을때 연결되는 url 표시
        - 제목: 사용자가 직접 입력
        - ip address: qr 스캔하였을때 연결되는 url의 iP 주소 표시
        - submit 버튼: 제목 사용자 저장용
            - sybmit 성공 → 팝업 → qr analysis history에 기록 추가
            - 실패시 → 팝업

▼ community 페이지

- community 글자: 누르면 가장 최근 게시글이 나와있는 페이지로 이동
- 게시글 목록 : 각 제목 클릭 시 게시글 세부사항 (제목, 내용, 사진, 댓글 등) 팝업 or 사이트 탭 형식으로 등장
- 게시글 세부사항 : 제목, 내용, 사진, URL, 댓글.
    - 댓글 : 최신 댓글 1개 표시 → 누르면 댓글 Drawer (내용 바로 하단까지) → 댓글 목록 및 댓글 작성 칸 & Submit 버튼 → 댓글 등록

▼ my post 페이지

- 게시글 작성 (윈)
    - 사진 첨부 : 장치에 저장된 사진 첨부 가능
    - 제목 : 제목 작성
    - URL : QR이 연결하는 URL

    - 내용 : 사용자가 다른 사용자와 공유할 정보 작성
    - Write 버튼 : 게시글 등록 완료
- My post 기록 (오)
    - my post 글자: my post 누르면 가장 최근 게시물이 나와있는 페이지로 이동
    - 게시글 목록 : 사용자가 작성한 모든 포스트 목록이 나타남(페이지네이션)

back

Java

▼ 헤더

로그인 시 토큰 발급

<로그인 상태 시>

- 회원 정보 수정 버튼 누르면 수정한 정보가 백엔드 → db 전달 및 수정 → 회원정보 수정 되었다는 응답을 프론트로
- 회원 탈퇴 버튼 누르면 백엔드 → db 전달 및 회원정보 삭제 →백엔드 토큰 삭제 → 삭제 완료 됐다는 응답을 프론트로

▼ 로그인 페이지

- 로그인 버튼 누르면 → 백엔드 → db 데이터 베이스의 아이디와 비밀번호가 저장된 번호와 일치하는지 비교 → 정보가 일치하면 백엔드 jwt 토큰 발급 → (성공/실패)응답을 프론트에 전달
- 구글 로그인 버튼 누르면 → (구글로그인하면 프론트에 발급되는 인가 코드) 백엔드에 인가코드 전달 → 백엔드에서 자신의 client id, client secret과 함께 구글 인증서버로 보냄 → (구글이 정보 유효성 판단하고) 구글에서 엑세스토큰과 사용자 정보 받아옴 → 엑세스 토큰 이용해 기존 회원은 로그인 처리, 신규 회원은 자동 회원가입 및 로그인 처리 → Jwt 토큰 발급 → (성공/실패)응답을 프론트에 전달
- 카카오톡 로그인 방식도 구글 로그인 방식과 유사

▼ 회원가입 페이지

- cofirm password 버튼 누르면 → DB 조회 → 사용 가능 여부 응답 → 프론트로 전달
- 회원가입 성공 : 성공 알림 응답 → sign up 버튼을 누르면 입력한 정보가 백엔드 → db 전달(비밀번호는 암호화), DB에 사용자 정보 저장 → 로그인 처리 → Jwt 토큰 발급 → 성공 응답 프론트에 전달
- 회원가입 실패 : 실패 알림 응답

▼ analysis 페이지(qr scan전 → analysis qr → qr scan 후)

- anaysis 글자 클릭 → 토큰 유효한지 확인
    - (토큰 유효)→ 성공 응답 프론트에 전달
    - (토큰 비유효) → 실패 응답 프론트에 전달
- qr scan 아이콘 탭
    - qr 스캔 아이콘:
        - 프론트에서 이미지 수신 → 스프링 → db저장& python 전달→ python에서 피싱 판별→ 분석 결과 DB에 저장 → 프론트로 전달 → 왼쪽에 QR analysis result 페이지 표시
- qr analysis history: 그동안 피싱여부 판별해 보았던 기록 확인
    - 이전 분석 기록 DB 조회 → 결과 반환
    - title 클릭: 해당 QR 분석 결과 DB 조회 → 결과 반환 → 왼쪽에 QR analysis result 페이지 표시
    - 페이지네이션 : DB 조회: **전체 데이터 중 11번째부터 20번째까지의 데이터만 가져오도록 쿼리**를 실행 →결과 반환
- qr anaysis result
    - 제목 넣고버튼 → 연결된 피싱 anlysis db 기록에 제목 추가 → 결과 응답 프론트에 전달

▼ community 페이지

- 게시글 목록 : 각 게시글 제목 클릭 → DB 조회 → 해당 게시글 세부사항 반환 → 프론트에서 형식에 따라 게시글 표시
- 게시글 세부사항 : 제목, 내용, 사진, URL, 댓글을 백엔드에서 DB 조회 → 결과 반환
    - 최신 댓글 : DB에서 가장 최근에 생성된 댓글 가져옴 → 프론트에 전달
    - 댓글 Drawer: 최신 댓글 클릭 → DB 조회 → 전체 댓글 목록 반환
    - 댓글 작성 : submit 버튼 클릭 → DB에 저장 → 저장 성공 시 성공 응답 → 프론트에서 목록 갱신
    - 페이지네이션 : DB 조회: **전체 데이터 중 11번째부터 20번째까지의 데이터만 가져오도록 쿼리**를 실행 → 결과 반환

▼ my post 페이지

- 게시글 작성 (왼)
  - Write 버튼 → 게시글 내용 db에 저장 (→ 제목과 내용 적지 않으면 오류 응답)→ 응답 반환
- My post 기록 (오)
  - my post 글자: my post 누르면 가장 최근 게시물이 나와있는 페이지로 이동
  - 게시글 목록: db 조회: 사용자가 작성한 게시글 select문 → 결과 반환
  - 페이지네이션 : DB 조회: **전체 데이터 중 11번째부터 20번째까지의 데이터만 가져오도록 쿼리**를 실행 →결과 반환

python

back

3

5. 데이터베이스 스키마


ANALYSIS
ANALYSIS_ID int NN
USER_ID varchar(45) NN
SCAN_RESULT varchar(25) NN
ANALYSIS_URL varchar(2083) NN
ANALYSIS_TITLE varchar(255) NN
IP_ADDRESS varchar(45) NN
created_at TIMESTAMP
updated_at TIMESTAMP
USER
USER_ID varchar(45) NN
USER_NAME varchar(45) NN
USER_PW varchar(255) NN
PHONE VARCHAR(20) NN
BOARD
BOARD_ID int NN
USER_ID varchar(45) NN
POST_TITLE VARCHAR(255) NN
IMAGE_PATH VARCHAR(255)
URL VARCHAR(2083)
POST_CONTENTS TEXT NN
created_at TIMESTAMP
updated_at TIMESTAMP
COMMENT
COMMENT_ID int NN
BOARD_ID int NN
USER_ID varchar(45) NN
COMMENT_CONTENTS TEXT NN
created_at TIMESTAMP
updated_at TIMESTAMP

6. 회의록

회의록

▼ 25.09.12
- 스프링 개념 15쪽씩 정리해서 수연이에게 넘기기
  - 수연이는 ppt 제작
- 각자 설치 중 문제점 보내기

▼ 25.09.15
- 에자일 프로젝트 관련 서비스 찾아보고 어떤 서비스 이용할지 결정
- UI 디자인 대략 그려보기

▼ 25.09.16
- Jira 사용법 공부하기
- figma 생성
- ▼ Jira

  create→에픽
  - task: 할 일
  - Feature: 기능 추가(요청)
  - Story: 기능 설명
  - Bug: 버그
  - Request: 지원 요청

▼ 25.09.17
-

회의록 1

▼ 25.09.19
- github organization 만들기

▼ 25.09.21

기능적 요구사항

▼ 25.09.17.(수)
- **피그마 UI 디자인 기간 설정(~09.21.(일))**
- **IntelliJ IDEA 설치 예정**
  - 스프링 부트 개발에 최적화된 기능을 제공하고, 코드 자동 완성과 오류 감지 등 학습과 협업에 유리하다는 점에서 IntelliJ IDEA를 개발 도구로 선정하기로 함.
- **UI 하위 항목 작성(최수연)**
  - 현재 디자인 계획안 외 추가로 더 제작해야할 페이지 구상하여 기록.
- **UI 디자인 완료 후 DB 설계**
  - UI에서 사용자가 입력하고 조회하는 데이터 구조를 명확히 파악하고, 그에 맞는 DB 구조 설계.
- **RAG에 대해 간단히 공부, BACKEND 기능 정리**
  - QR 피싱 사이트에서 RAG을 활용하기로 결정하였고, 이를 위해 RAG에 대해서 조사해 보기로 결정.
  - UI 디자인 후 프론트와 연결되어질 BACK 기능을 정리 해 추후 개발이 계획적으로 이루어지도록 할 것임.
- **로고 및 마스코트 제작**
  - 생성형 AI를 활용하여 로고를 제작하였고, 후보군 3개를 추려냈음.
  - 각각 3개를 figma에 삽입해보고 UI와 어울리는 로고를 택하기로 결정.
  - ▼ 로고 후보군

회의록 2

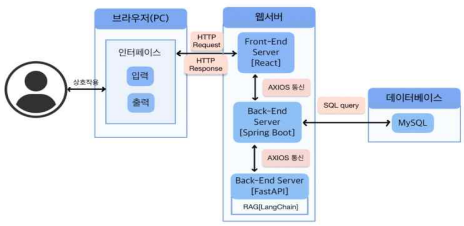
▼ 25.09.19.(금)
- **IntelliJ IDEA 사용 방법 및 shadcn 컴포넌트 프론트에 적용하는 방법 조사(심연우)**

회의록 3

- **DB 구성 계획(최수연, 김여민)**

▼ 25.09.20(토)
- figma 리뷰 및 수정사항 정리, 수정
- 과제 제출 서류 작성
- github와 jira 연동 방법 공부(https://lamerry.tistory.com/entry/Jira-Github-연동하기-1-Jira란?category=1265013)

▼ 25.09.21(일)
- 기능적 요구사항 작성(백엔드/프론트엔드)
- 시스템 정의서 최종본 작성



- function 수정

▼ 25.09.22(월)
- shadcn 컴포넌트 적용 방법 알기
- react 1차 세팅

▼ 25.09.23(화)
- intellij 버전 설정 및 환경 세팅

회의록 4

- figma 리뷰 및 수정

▼ 25.09.25(목)

- spring 대부분의 기능 추가 완료
- api 검사 완료
- figma 수정()

▼ 25.09.26(금)

- spring 기능 추가(게시글 및 댓글 삭제와 신고)
- api 추가 검사
- figma 수정(게시글 신고)
- 로고 및 마스코트 결정

의의록 5

7. Figma UI 디자인