

우아한형제들

테스트 엔지니어

사전 과제 수행 보고서

김나예

문제 1-1

주어진 API 명세에 맞춰 메뉴선택 - 주문생성 시스템의 핵심 기능을 검증하는 테스트 자동화 코드 구현

아래 Git 저장소에 테스트 자동화 구현 전체 코드 및 문서 산출물을 업로드 했습니다.

- <https://github.com/nayekim96/API-Test-Automation>
- 테스트 가이드는 [README.md](#) 파일을 참고하시고, docs 디렉토리에서 문서 산출물 확인 가능합니다.

문제 1-2

실제 서비스 운영 단계에서 발생할 수 있는 다양한 오류 상황과 예외 케이스 및 해결 과정 기술

- 오류 및 예외 케이스 항목과 해결과정을 표로 작성했습니다.

종류	분류	오류 상황 및 예외 케이스	해결 과정
환경	서버	- 트래픽 증가로 운영 서버가 다운되는 경우 - 클라우드 인프라 장애 발생하는 경우	- 앱 UI 및 기본 기능 검증을 자동화 하여 빈 화면이 노출되는지 수시로 모니터링하여 즉각 조치 지원 - 핵심 기능에 대한 API 검증 자동화로 정상 케이스에서 에러가 발생하는지 수시로 체크 - 사용하는 API Endpoint에 대한 Health Check 모니터링으로 서버 상태를 미리 확인하여 조기 대응
	DB	- DB 서버가 다운되는 경우 - DB 접근 권한이 만료되어 접근 차단되는 경우 (회원정보 조회 불가)	- DB 연결 및 권한, 실제 데이터 접근에 대한 Health Check를 정기적으로 진행해 조기 대응 지원

호환성	디바이스	<ul style="list-style-type: none"> - 모바일 앱의 경우 디바이스 OS 버전 혹은 디바이스 종류에 따라 호환성 에러가 발생할 수 있음 - 디바이스 해상도에 따라 UI가 깨지는 현상이 발생할 수 있음 	<ul style="list-style-type: none"> - 다양한 종류의 디바이스와 OS, OS 버전에서 기능 테스트 진행
	앱 버전	<ul style="list-style-type: none"> - 하위 버전 앱에서 신규 기능 등이 실행되지 않을 수 있음 	<ul style="list-style-type: none"> - 배포 전 빌드 검증 기간 동안 최근 5개의 버전에서 기능 테스트 진행
	URL Scheme	<ul style="list-style-type: none"> - 잘못된 URL 스킴으로 인한 오류 발생 가능 	<ul style="list-style-type: none"> - URL Scheme 검증을 자동화 하여 정기적으로 앱연동 확인
보안	민감정보 노출 및 SQL Injection	<ul style="list-style-type: none"> - API 호출 시 민감정보가 유출되거나 SQL Injection 등 보안 에러가 발생할 수 있음 	<ul style="list-style-type: none"> - 모든 API 입력값에 대해 유효성 조건을 설정 및 검증 수시 진행 - API 호출 로그를 분석해 수상한 입력값이나 SQL 오류 메시지 발생 검증 수시 진행
성능	응답 시간	<ul style="list-style-type: none"> - 앱 실행 혹은 기능 수행 시 응답 속도가 느려지는 상황 발생 가능 	<ul style="list-style-type: none"> - 화면 전환 속도, 배터리 소모량 등 성능 측정을 자동화 하고 모니터링 시스템(시각화)구현으로 화면 전환 속도 이상을 조기에 감지
	배터리 소모량	<ul style="list-style-type: none"> - 앱 사용 시 배터리가 과다 소모되는 상황이 발생할 수 있음 	

문제 2

결함 수정 비용을 감소시키기 위하여 필요한 활동을 간략히 기술

과제 문서에 명시된 그래프는 개발 초기에 감지할 수 있는 결함 비율이 가장 높으나, 결함 수정 비용은 개발 초기가 가장 저렴함을 나타내고 있습니다. 반대로, 배포 이후의 이슈는 감지하기 힘들며 수정 비용이 매우 높습니다.

따라서 초기에 결함을 최대한 많이 발견하고 수정하는 것이 중요한데, 이를 위해 **조기 테스트**가 필요하다고 생각합니다.

예를 들어, 개발 프로세스 초기 단계부터 QA 활동을 하여 요구사항 및 설계 문서, 기획서 검토 등 정적 분석을 진행해 설계 상의 문제를 사전에 방지할 수 있습니다. 또한, DevOps에 SonarQube와 같은 정적 코드 분석 툴을 사용해 코드 품질 검증을 자동화 할 수 있습니다. 이 활동으로 코드 통합 단계에서 버그나 보안 취약점을 감지해 결함을 예방할 수 있습니다.

이렇게 QA 엔지니어도 개발 초기에 코드 리뷰나 정적 분석 활동에 참여해야 하기 때문에, SW 개발 관련 지식과 신기술을 지속적으로 학습해 역량을 쌓는 것 또한 중요하다고 생각합니다. 코드 이해력과 분석력이 높을 수록 코딩 단계에서 많은 결함을 발견할 수 있을 것입니다.