

이 조사결과에 따르면 가장 많은 취약점이 발견된 언어는 C언어였다. 오픈소스는 프로그래밍시 구글 혹은 다른 사이트를 통해서 참고할 수 있는 중요한 자료이다. 하지만 시큐어코딩을 거치지 않고 바로 사용하게 될 경우 시스템에 치명적인 취약점을 입힐 가능성이 있다.

본 논문에서 설계하고 구현한 모바일 애플리케이션은 개발자들에게 시큐어 코딩 개념 원리에 대한 설명을 제공하고, 객관식 문제를 풀어보며 취약점에 대한 정보를 인식할 수 있도록 하고 있다. 문제에 대해서 완벽하게 이해했을 때 사용자에게 인증서를 제공하며 시큐어 코딩에 대한 자신의 이해도를 점검하는 방식을 도입했다. 실습 문제를 통해서 안전하지 않은 코드를 안전한 코드로 바꾸는 방향성을 제시하며 결과만을 중요시하는 코딩이 아니라 프로그래밍을 안전하게 구현하며 개발자가 본인의 코드를 이해하는 것이 가장 이상적인 코딩임을 인식할 수 있도록 목적을 두고 있다.

현재 앱스토어 시장에서는 코딩 교육에 대한 앱들은 많지만 시큐어 코딩을 교육하는 앱은 설계되어 있지 않다. 또한 시큐어 코딩 방법을 습득할 수 있는 교육기관이 부족해 기술 습득 기회가 많지 않은 것이 사실이다[4]. 앞으로도 소프트웨어 개발보안과 시큐어 코딩 적용의 중요성은 증가할 것이다[5].

본 애플리케이션은 KISA에서 제공하는 C 시큐어 코딩 가이드를 참고하여 앱으로 만들어 실용성을 높였다. 웹 서비스와 비교했을 때 실제 코딩을 하는 창을 블록으로 구현하였기 때문에 프로그래밍할 때도 보다 편리하게 사용할 수 있도록 기획하였다.

2. 관련 연구 및 관련 기술

2.1 시큐어 코딩과 취약점

소프트웨어를 개발하기 위한 과정에서 프로그래머가 코딩한 프로그램에 많은 오류가 발생한다. 이러한 오류는 검증에 의해 발견이 되지만, 발견되지 않은 오류가 존재하며, 프로그램을 사용하는 중에 발견되는 경우가 있다. 이러한 경우에는 수정에 많은 시간과 비용이 소모된다. 이러한 오류는 사용 중인 프로그램이 내재한 기능성의 문제를 발생시킬 뿐만이 아닌 보안성에 중요한 문제인 프로그램 취약성을 발생시킨다[6]. 위 문제를 해결할 수 있는 것이 시큐어 코딩이다. 시큐어 코딩은 소프트웨어 개발 단계(Software Development Life Cycle : SDLC)에서 보안 약점을 제거함으로써 소프트웨어의 취약점과 해킹의 위험성을 줄여주는 방어적 프로그래밍 기법이다[7]. 시큐어 코딩

은 입력데이터의 검증 및 표현, 보안 기능, 시간 및 상태, 오류 처리, 코드 오류, 캡슐화, API 오용 등 다양한 취약점에 대응할 수 있게 되어 있지만, 주로 웹 서버 혹은 웹 애플리케이션이 직접적으로 가지는 취약점에 대해 조치하는 것이 일반적이다[8]. 특히, 소스코드 보안취약점을 이용한 사이버 공격은 침입차단 및 침입방지 시스템 등 일반적인 보안 장비로는 대응이 어려운 특징이 있다[9].

2.2 사용한 개발 프레임 워크 Flutter

플러터는 한 번 코딩으로 여러 플랫폼용 앱을 만드는 크로스 플랫폼 개발 프레임워크다. 안드로이드와 iOS뿐 아니라 웹, 데스크 톱, 앱 개발도 가능하다. 장점으로 낮은 진입장벽, 높은 네이티브 성능, 훌륭한 개발 도구지원을 들 수 있다[10]. 플러터 언어인 Dart를 사용할 경우 iOS와 안드로이드를 모두 호환하는 앱을 개발할 수 있다.

플러터의 주요 구성요소는 다음과 같다:

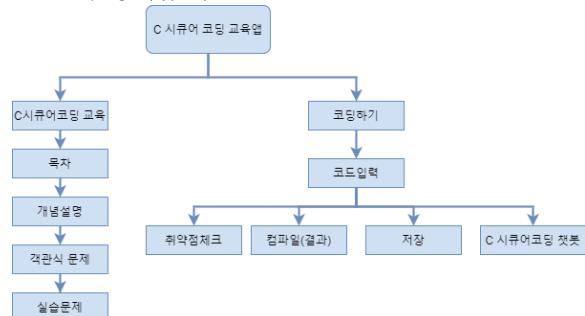
- 다트 플랫폼
- 플러터 엔진
- 파운데이션 라이브러리
- 디자인 특화 위젯 [11]

본 애플리케이션 앱을 설계하기 위해서 하나의 언어로 안드로이드와 iOS 운영체제를 모두 포괄할 수 있는 flutter를 선택하였다.

3. 설계 및 구현

3.1 구조적 설계

본 애플리케이션은 C 시큐어 코딩 교육과 코딩하기를 주요 기능으로 초점을 두고 있다. 그림 3은 애플리케이션의 전체 시스템 구성도로써 인터페이스별 핵심 기능을 중심으로 구성하였다.



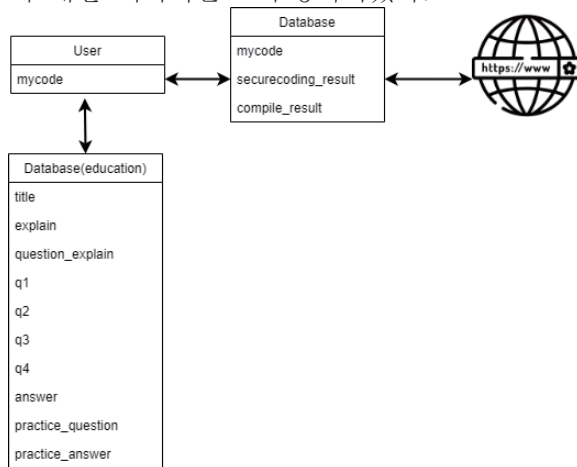
[그림 3] 앱의 시스템 구조도

아래 그림 4는 데이터베이스 구조도를 보여준다. 이 중에서 'Secure Studying'은 User, Database, Database(education)로 설계되어있다.

User는 입력한 mycode를 database로 보내 저장을 하고 해당 코드를 컴파일한 결과들

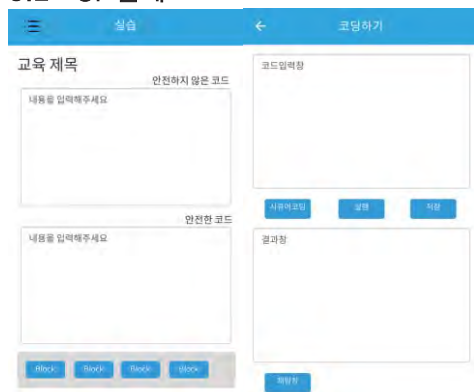
을 securecoding_result와 compile_result에 크롤링하여 저장한다.

Databases(education)은 C 시큐어 코딩 교육에 필요한 개념설명, 개념문제, 실습문제에 대한 데이터들로 구성되어있다.



[그림 4] 데이터베이스 구조도

3.2 UI 설계



[그림5] 교육하기 실습 문제 UI

[그림6] 코딩하기 실행 UI

그림 5와 6은 ‘Secure Studying’의 사용자 인터페이스를 나타낸 그림이다. 코딩하기와 실습하기의 기본적인 기능을 화면에 보여준다.



[그림 7] OWASP 표준 2017 빈도수가 큰 취약점 TOP 10

메인페이지에서는 로고 레이블과 코딩하기, 실습하기 버튼 2개를 가진 기본적인 인터페이스라 따로 그림을 첨부하지 않았다. 실습하기 창에서는 취약점 별로 문제와 개념을 정리해 놓았으며 문제는 그림 7에 나와 있는 OWASP에서 정의한 빈도수 별 취약점을 토대로 정리하였다.

코딩하기는 사용자가 코드를 입력할 코딩 창, 결과를 보여주는 결과 창을 토대로 구성하였다. 시큐어 코딩, 결과, 저장하는 버튼을 코딩 창과 결과 창 사이에 배치하였다.

3.3 구현

본 어플리케이션 ‘Secure Studying’은 별도의 로그인 없이 사용 가능하다. 메인 화면에서 ‘C 시큐어 코딩 교육’ 버튼과 ‘코딩하기’ 버튼 중 원하는 버튼을 클릭하면 해당 인터페이스로 이동한다. ‘시큐어 코딩 교육’을 클릭하면 교육 목차들이 나열되어 있고 원하는 목차를 선택하면 해당 목차에 대한 개념 설명을 볼 수 있다. 문제는 객관식과 실습 문제로 구성되어 있다. 실습 문제는 블록코딩으로 이루어진다. 잘못된 코드 부분을 보여주고 빈칸으로 둔다. 해당 코드를 블록으로 표시하고 사용자는 블록을 드래그 하여 알맞게 배치하면 된다. 정답에 맞게 배치하면 자동으로 넘어간다. ‘코딩하기’ 버튼을 클릭하면 ‘코드 입력 창’과 ‘시큐어 코딩’, ‘실행’, ‘저장’, ‘채팅 창’으로 이루어진 인터페이스로 이동한다. ‘시큐어 코딩’은 입력된 코드에 대한 보안적 문제들을 cppCheck 사이트에서 웹 크롤링을 통해 결과 창에 출력하여 개발자가 보안적인 취약점에 대해 인식할 수 있도록 한다. ‘실행’은 사용자가 입력한 코드를 서버가 컴파일러 사이트에 전송한다. 컴파일된 결과를 웹 크롤링을 통해 ‘결과 창’에 출력한다. ‘채팅 창’은 컴파일 결과에 오류가 출력되었을 때, 해당 오류를 복사하여 채팅 창에 붙여넣으면 문제 해결에 도움을 주는 웹사이트들의 URL을 보여준다.

4. 결론

본 논문에서는 개발자들이 실생활에서 시큐어 코딩에 대한 이해도를 높이고 교육을 더욱 쉽고 간편하게 이용할 수 있도록 블록 형태의 교육 애플리케이션을 설계하고 구현하였다. 본 논문에서 구현한 애플리케이션은 웹 서비스보다는 일상생활에서 간편하게 사용하고 교육할 수 있도록 스마트 폰에 초점을 두어 모바일 애플리케이션으로 구현하여 사용자의 편의성과 학습효과를 높일 수 있을 것으로 기대된다.

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학지원사업의 연구결과로 수행되었음(2016-0-00022)

참고문헌

- [1] 2019 년 대학 SW 특기자 전형 사진
<https://blog.naver.com/wlrhks815/221876328288>
- [2] LG CNS 공식 블로그
<https://m.post.naver.com/viewer/postView.nhn?volumeNo=16691275&memberNo=3185448&vType=VERTICAL>
- [3] 최문정, 최준성, 정익래, “시큐어 코딩 관련 국내 연구 동향”, 한국통신학회 학술대회논문집, 한국통신학회, 2015, 790-791
- 사진[2] <https://byline.network/2019/03/27-42/>
- [4] “파수닷컴, SW 개발자 대상 시큐어코딩 교육”, 디지털 타임즈, 2014.07.11,
http://www.dt.co.kr/contents.html?article_no=2014071102109960800002
- [5] 최문정, 최준성, 정익래, “시큐어 코딩 관련 국내 연구 동향”, 한국통신학회 학술대회논문집, 한국통신학회, 2015, 790-791
- [6] 김슬기, 박대우, SW 취약점의 보안성 강화를 위한 진단원의 교육 양성 연구, 한국정보통신 학회논문지, 한국정보통신학회, 2017, 945-950
- [7] A. John. R. Peter, “Electric Communication Development,” Communications of the ACM, 40, PP. 71-79, May. 1997.
- [8] S. M . C ho, H. L ee, “A Countermeasure against the Abatement Attack to the Security Server,” Journal of the Korea Institute of Information and Communication Engineering, vol. 20, no. 1, pp. 94-102, Jan. 2016
- [9] 김슬기, 박대우, SW 취약점의 보안성 강화를 위한 진단원의 교육 양성 연구, 한국정보통신 학회논문지, 한국정보통신학회, 2017, 945-950
- [10] 오준석, “오준석의 플러터 생존코딩”, 한빛 미디어, p25
- [11] <https://ko.wikipedia.org/wiki/%ED%94%8C%EB%9F%AC%ED%84%B0>