

S.O.S (소셜 온-오프네트워크 서비스)

요 약

온라인에서 뿐만 아니라 오프라인을 통해서도 사회적 관계망을 형성할 수 있는 S.O.S 를 개발하고자 한다. 이는 기존 SNS 들이 가진 한계점들을 보완하고 장점을 흡수하여 건전한 사회적 관계를 만들기 위한 어플리케이션이다.

1. 서론

1.1. 연구배경

대부분의 사용자들은 온라인 상에서 뿐만 아니라 오프라인상에서도 다양한 사회적 관계형성에 대한 기대를 원하고 있다. 그리고 기존 SNS 에서는 본인과 비슷한 의견과 생각 등을 지속적으로 수집함으로써 편향된 사고를 가질 확률이 매우 높다. 이로 인해 다양한 사람들과 갈등이 생길 확률이 매우 높다. 결과적으로 서로에게 도움을 주고 받는 것에 대해 각박한 사회가 되어가고 있다. 따라서 스타트업 드림의 일환으로 S.O.S 라는 어플리케이션을 개발하여 많은 도움들을 주고 받을 수 있도록 하여 온라인 뿐만 아니라 오프라인에서도 사회적 관계를 만들어 건강한 사회를 만들고자 한다.

1.2. 연구목표

SOS(소셜 온-오프라인 네트워크 서비스)는 온라인상에서 외에도 오프라인에서도 다양한 사회적 관계를 형성하여 서로에게 도움을 주고 받는 것에 각박한 사회에서 벗어나고자 한다. 또한, 기존의 SNS 로 인한 갈등 및 중독 등의 사회적 문제를 해결하고자 한다. 그리고 기존 SNS 에서 본인과 비슷한 의견들을 수집하여 편향적인 사고를 가지는 것에서 벗어나 다양한 생각과 가치관의 콘텐츠를 공유할 수 있는 형태의 SNS 를 개발하고자 한다.

2. 관련연구

2.1. Kotlin

2.1.1. Java 와 Kotlin 의 비교

Java 와 Kotlin 은 모두 정적타입으로 컴파일러에서 Type-Error 를 찾을 수 있습니다. 또한, 멀티 플랫폼 개발을 할 수 있는 장점을 보유하고 있습니다. 그리고 모두 객체지향 프로그래밍을 지원하여 class 키워드를 사용할 수 있는 공통점을 가지고 있습니다.

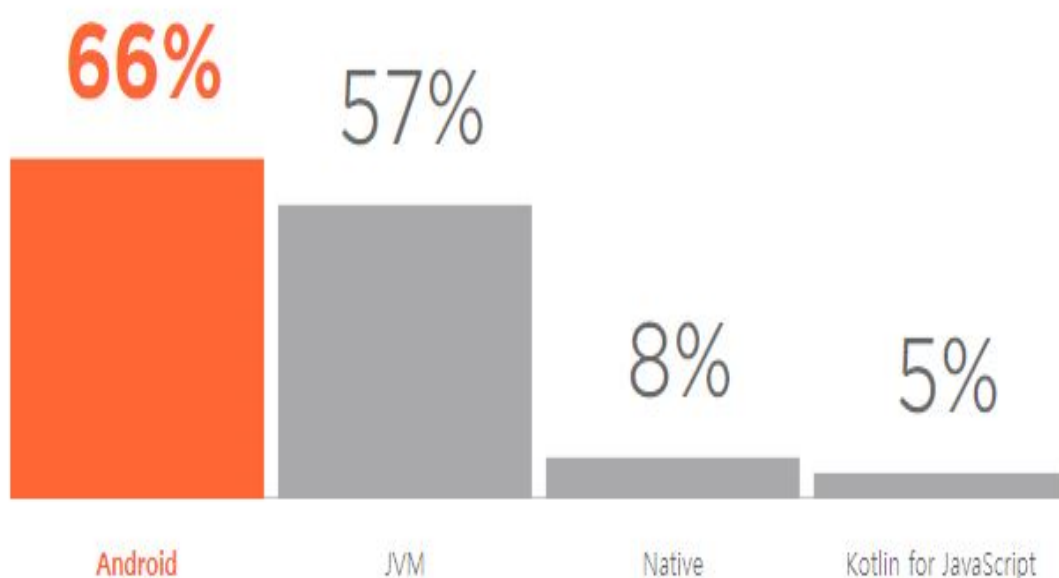
하지만, Java 는 class 가 기본단위로 class 안에서 로직을 만들어야 하는 반면, 코틀린은 class 가 선택사항입니다. 따라서 Top-Level 에서 변수, 함수의 선언과 구현이 가능합니다. 그리고 자바 9 이하의 버전에서는 타입 추론이 불가능하지만, 코틀린은 타입추론이 가능합니다. 마지막으로 Null Point 의 예외처리에서 자바의 경우에는 런타임에 Null 참조시 예외를 발생시킬 수 있습니다. 그에 반해 코틀린은 Null 을 확인하는 연산자가 존재하여 이를 컴파일 시점에서 방지할 수 있습니다.

	Type system	Multi-Platform	OOP	FP	Type Inference	Nullable
자바 (Java)	정적 타입	0	0	X	X (9 이하)	X
코틀린 (Kotlin)	정적 타입	0	0	0	0	0

[그림 1] 자바와 Kotlin 비교 표

2.1.2. Kotlin 의 장점

코틀린은 다른 사람이 이해하기 쉽도록 코드를 작성할 때 시간이 단축된다. 이로 인해 코드가 짧으며 가독성이 매우 좋다. 이로 인해 에러발생률을 낮출수 있다. 또한, coroutines, extension functons, lambdas 등의 다른 라이브러리와 Android Jetpack 을 지원한다. 그리고 Java 와 호환이 가능하여 기존의 어플리케이션을 모두 Kotlin 으로 손쉽게 바꿀수 있다. 이로 인해 Kotlin 은 2019 년 기준으로 안드로이드나 JVM 에서 대부분 사용하고 있다.



[그림 2] 2019 년 기준 개발자들의 Kotlin 을 언어로 하는 플랫폼 비중

2.2. Spring Boot

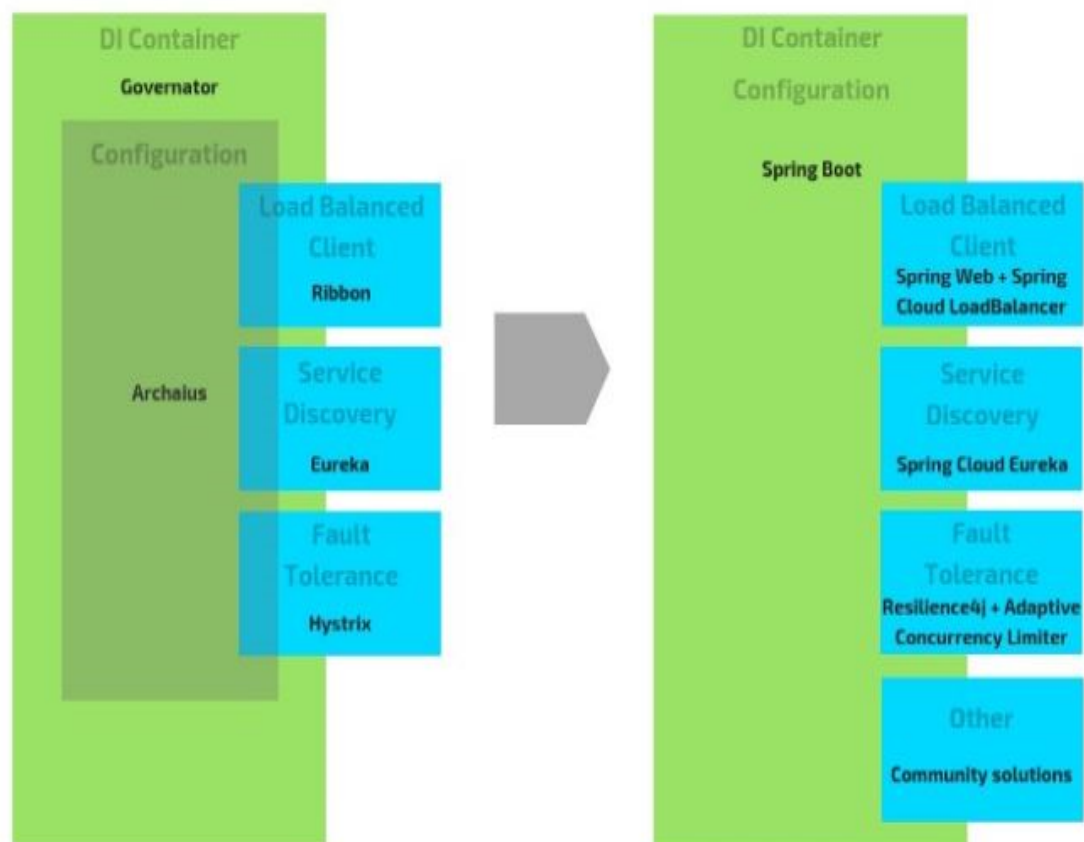
Spring Boot 는 스프링 포트폴리오를 신속하게, 미리 정의된 방식으로, 이식성 있게, 실제 서비스 환경에 사용할 수 있도록 조립해 놓은 것이다.

신속성: 의존관계를 포함하여 애플리케이션의 여러 요소에 기반한 의사결정을 신속하게 적용할 수 있게 하여 개발속도를 높인다.

미리 정의된 방식: 스프링 부트의 구성을 정하면, 그 구성에 따라 미리 정의된 방식으로 기본적인 설정을 자동으로 지정해 준다.

이식성: 자바의 표준도구인 스프링 프레임워크를 기반으로 만들어져, JDK 가 있다면 어느 환경에서나 실행할 수 있다. 또한, 특정한 인증을 받은 애플리케이션 서버가 필요하지 않으므로 스프링 부트로 애플리케이션을 만들고 패키지를 만들면 어디에서든 배포하여 실행이 가능하다.

실제 서비스 환경에서 사용 가능: 작은 부분에서 사용해야 하는 제약이 없으며, 실제로도 광범위하게 사용되고 있다. 실제로 넷플릭스의 경우에는 클라우드 인프라의 요구사항인 안정성, 확장성, 효율성, 보안을 충족하기 위해 스프링 부트를 채택하였고, 일부는 넷플릭스 자체의 소프트웨어의 사용 및 조정을 통해 인프라를 발전시켰습니다.



[그림 3] Netflix 의 Spring Boot 사용 사례

3. 프로젝트 내용

3.1. 마이 SOS

- 마이 SOS 는 크게 2 개의 화면으로 구성된다.
 - 사용자가 작성한 SOS 목록
 - 사용자가 도와주려는 사람들의 SOS 목록
 - 도움을 받기 위한 SOS 등록
- 사용자가 작성한 SOS 목록에는 다음과 같은 내용이 표시된다.
 - 도움 내용, 위치 및 비용
 - 도움의 수락 여부 (수락 또는 거절)
- 사용자가 도와주려는 사람들의 SOS 목록은 다음을 포함한다.
 - 도움 내용, 위치 및 비용
 - 수락 여부 (승인 시: 메시지, 거절 시: 죄송합니다, 처리 X: 대기중)
- SOS 를 등록하기 위해서는 다음과 같은 내용을 입력하도록 한다.
 - 요청 사항에 대한 글
 - 위치 및 비용
 - 사진 및 동영상 (선택)

3.2. 메시지

- 메시지 기능은 1:1 채팅 형식으로 구현하도록 한다
- 메시지는 글 작성 외에 사진 또는 동영상도 보낼 수 있도록 허용한다.
- 1 대 다 채팅을 위한 초대 기능은 현재는 구현하지 않고 추후에 구현하도록 한다.

3.3. 마이페이지

- 마이페이지에서 보여줄 내용은 다음과 같다.
 - 사용자의 프로필 동영상 또는 사진
 - 사용자가 올린 글
- 마이페이지에 들어갈 화면으로는 마이페이지 메인 화면, 프로필 수정 화면이 존재한다.

4. 향후 일정

일정	내용
9.11 ~ 9.30	요구사항 분석 및 기능정리
10.1 ~ 11.15	시스템 설계 및 구현
11.16 ~ 11.30	테스트 및 예외처리
12.1 ~ 12.18	최종 완성 및 최종발표.보고서 작성

5. 결론 및 기대효과

SOS는 서로 돕고 다양한 가치관이 공존하는 사회를 추구하고, 온-오프라인의 복잡한 사회적 관계망 서비스를 지향하고 있다. 또한, 도움 플랫폼을 통해 팀원 모집, 중고 거래 등의 다양한 서비스들을 중개함으로써 행복한 사회를 꿈꿀 수 있을 것으로 기대된다. 그리고 차후에는 온라인 의류 쇼핑몰이나 맛집 등과 협업을 통해 서비스를 확장하여 다양한 반응형.노출형 광고를 통한 수익을 얻을 수 있을 것으로 예상된다.

6. 참고문헌

- [1] 스프링 부트와 AWS로 혼자 구현하는 웹 서비스
- [2] 스프링 부트 실전 활용 마스터
- [3] [Android] 코틀린과 자바의 차이 <https://bbaktaeho-95.tistory.com/50>
- [4] Kotlin 2019 - 2019년 개발자 에코시스템의 현황 인포그래픽
<https://www.jetbrains.com/ko-kr/lp/devecosystem-2019/kotlin/>
- [5] Kotlin 공식 문서 <https://kotlinlang.org/docs>
- [6] Netflix의 스프링 부트 사용 사례
<https://netflixtechblog.com/netflix-oss-and-spring-boot-coming-full-circle-4855947713a0>