

프로젝트 최종 보고서

1. 주제

긴급신고 시스템을 위한 중복 신고 알림 시스템 개발

2. 팀원: 나반 14 팀

20221819 한준협 (팀장), 20221806 이정원, 20223102 최주영, 20223081 신준철

3. 요약

119 혹은 112 에 대한 긴급 신고 시스템을 기반으로, 비슷한 유형의 신고가 일정 거리 안에서 다수 발생하게 되면 시스템에서 이를 파악하여 재난 대응 단계 격상을 제안한다. 프로젝트의 예상 이용자는 소방서에서 신고를 처리하는 파트에서 일하시는 분들이 대상이다. 신고가 들어오면, 담당자가 신고 접수용 페이지에서 신고를 접수한다. 서버에서는 신고를 데이터베이스에 저장함과 동시에 가장 최근 신고 반경 3 km 내에 비슷한 유형의 신고가 다수 존재하는지 판별 한 후 조회용 페이지에 알림을 보내준다.

현재 소방 신고 체계는 접수된 신고에 대해서 비슷한 신고가 들어온 횟수를 안내하지는 않는다. 단지 긴급신고전화를 받는 직원분께서 직접 확인을 하고 대응 단계를 격상할지 판단한다. 하지만 사람이 항상 모니터링을 할 수 없고, 많은 직원들이 출동을 나간 상태에서 인력이 부족하다면 판단을 내릴 여력이 없을 수 있다. 이 프로젝트는 이 부분에서 도움을 줄 것이다. 접수된 신고에 대해서 서버가 모니터링 하여 일정 시간 동안 비슷한 유형의 신고에 대해서 파악을 한뒤, 관리자에게 격상을 할 것을 알림으로 보내줌으로 인해 재난 대응 시간을 줄일 수 있다.

4. 개발 배경 및 예상 결과

22 년 할로윈 축제가 한창일 때, 이태원에서 수많은 사람들이 압사로 세상을 뜨는 사고가 발생했다. 사고 당시 112 와 119 에 신고가 여러 차례 들어갔지만 당국은 제대로된 대응을 하지 못하였다. 이번 사고를 계기로 기존 긴급 신고 시스템에 부족한 점이 있다고 판단하여 본 프로젝트 개발을 시작하였다.

본 프로젝트를 실제 긴급 신고 시스템에 적용한다면 현재 긴급신고 통합서비스를 바탕으로 접수된 신고의 특징을 파악하고 재난이라 판단될 경우 상황실에 알림을 보내 빠른 초동대응이 가능하게 해준다. 만약 한 지역 관할 상황실에 사람이 부족하여 결정을 하지 못하는 상황일지라도, 다른 지역에서도 본 알람을 수신하여 대응을 할 수 있어 긴급신고 통합서비스를 개선할 수 있을 것으로 예상된다.

5. 서론

5-1. 배경

이태원은 매년 10 월 말이 되면 할로윈을 즐기는 사람들이 몰려드는 장소중 한 곳이다. 2020 년 웹툰과 드라마로 인기를 끈 ‘이태원 클라쓰’에서 이태원의 할로윈 축제는 젊은 세대의 자유를 상징하는 문화로 묘사되기 까지 했다. 2021 년에는 코로나로 통제가 심했던 시기임에도 하루 약 8 만 명 이상 인파가 몰렸을 정도였다. 2022 년 3 년만에 거리두기가 없이 축제가 진행되어 시민들이 10 만 명 이상 이태원으로 몰렸다. 22 년 10 월 29 일 밤, 많은 사람들이 거리로 나왔고 이태원에서 축제가 시작된 상황에서 많은 인파가 거리로 모였다. 많은 사람들이 경사진 좁은 골목에 갇혀 빠져나오지 못한 채 압사로 인해 사망하는 참사가 발생했다. 233 명 (10 월 30 일 오전 기준)의 사상자가 발생한 이번 참사는 소방서와 경찰의 대응이 늦었다는 의견이 다수 제기되었다. 11 월 8 일 경향신문의 한 기사에 따르면 소방서에서 경찰의 공조요청을 2 번 무시 하였다하고, YTN 의 11 월 2 일 기사에 따르면 ‘이태원 참사’ 112 신고 녹취록을 확인해본 결과 참사를 방치한것과 같았다고 보도하였다. 참사가 발생하기 전에 몇 차례의 신고가 접수되었지만 상황실에서는 재난이라 생각하지 않고 제대로된 대응을 하지 않은 것이다.

5-2. 문제정의

현재 112 나 119 와 같은 긴급 신고전화는 ‘긴급신고 통합시스템’으로 관리되고 있다. 어느 한쪽으로 신고를 하더라도 같은 전산망에 저장되어 따로 신고를 할 필요가 없는 것이다. 이 시스템은 경찰.소방.해경간 긴급신고에 대해 상호 공동대응이 필요한 경우 신고정보(신고 내용, 재난 위치, 신고자 연락처 등)를 공유한다. 각 기관의 상황실 직원들은 이 정보들을 바탕으로 재난 대응 단계 격상과 같은 판단들을 내리고 업무를 수행하게 된다. 이 부분에서 하나의 문제점이 발생한다. 앞서 언급한 재난 대응 단계 격상은 직원이 신고를 모니터링 하고 판단을 내리게 되는 것인데, 사람이 항상 주시하고 있을 수는 없고 인력이 부족할 경우 판단을 내릴 사람이 없을 수도 있다. 이번 이태원 참사에서는 인원 부족으로 인해 재난 대응 격상과 같은 판단을 내릴 사람이 부족했다.

5-3. 극복 방안

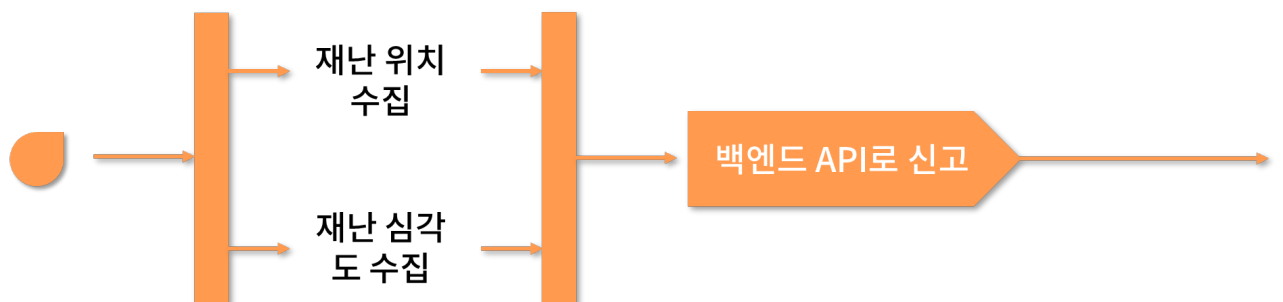
본 프로젝트는 문제정의에서 언급한 ‘재난 대응 단계 격상’에 초점을 두고 진행하였다. 긴급신고 통합시스템을 통해 공유받은 신고나 소방서에서 직접 접수한 신고에 대해서 신고를 접수한 직원이 신고의 정보(신고 내용, 신고자의 위치, 심각성, 등)를 서버로 전송한다. 서버에서는 최근 들어온 신고에 대해서 반경 3 km 이내의

신고들을 확인해본다. 각 신고들의 내용을 검사해 이번 신고와 유사성을 파악하고 유사도가 높은 신고들 개수를 계산한다. 만약 일정 수준 이상의 신고가 접수된 것을 확인하면 가장 최근 신고에 대한 재난 대응 단계의 격상을 제안한다. 서버에서는 이 제안을 소방서 근무자가 보는 페이지에 팝업 혹은 시인성이 좋은 경고페이지를 통해 알려준다. 이를 통해 근무자가 알림을 통해 일정 구역 내에서 유사한 신고가 여러 번 접수된 사실을 파악할 수 있고 재난 대응 단계를 격상 시킬 수 있다. 재난 대응 단계는 사람이 올리지만 격상 제안 알림은 다른 소방서에서 근무하는 직원들도 확인할 수 있다는 점에서 인력 부족으로 인해 생기는 문제는 피할 수 있게 된다.

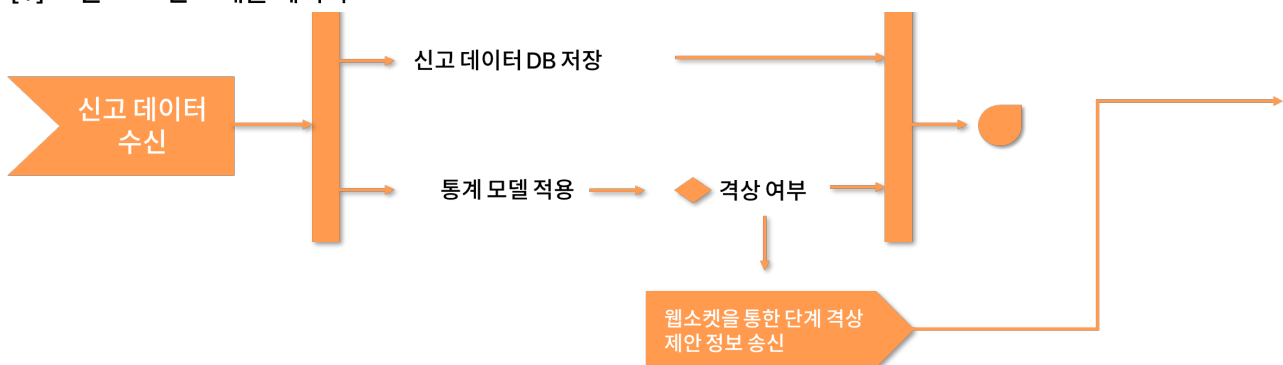
즉, 기존에는 재난 대응 단계 격상을 사람이 판단하고 현재 신고 상황을 항시 모니터링 하고 있다. 이 부분에서 본 프로젝트를 통해 반복되고 유사한 신고를 상기시킴으로 재난에 빠른 대응이 가능하고 프로그램으로 신고현황에 대한 모니터링을 자동화하여 얻는 이익이 많을 것으로 예상된다.

6. 본론

6-1. 시스템 개요 그림



[1] 프론트 - 신고 제출 페이지



[2] 백엔드 - 서버



[3] 프론트 - 신고 조회용 페이지

6-2. 시스템 구조

1. 프론트 엔드 - 신고 페이지

신고가 접수되면 신고에 대한 내용, 위치와 심각도에 대해서 정리를 한다. 만약 신고에 대해 특이사항 혹은 메모할 내용이 있다면 앞에서 얻은 데이터와 함께 신고 접수를 담당할 직원이 신고 페이지를 통해 백 엔드 API로 신고를 한다.

2. 백 엔드 - 대응 단계 격상 판단

신고 페이지를 통해 접수된 신고 데이터를 수신하여 '신고 데이터 DB'에 저장한다. 동시에 통계 모델에 적용하여 격상 여부를 판단하게 된다. 만약 시스템이 판단하기에 신고에 대해서 재난 대응 단계를 격상시켜야 한다고 판단된다면 웹소켓을 통한 단계 격상 제안 정보를 송신한다.

* 판단 과정 : 접수된 신고의 위치를 파악한 후, 반경 3km 내에 있는 유효 신고 기한 (3시간)이 지나지 않은 신고들을 선정한다. 이후 선정된 신고들의 내용과 접수된 신고의 내용을 비교하여 유사도를 검사한 다음, 유사도가 높은 신고들의 개수가 5개* 이상이 된다면 재난 대응 단계의 격상을 제안하게 된다. (* 임의로 지정한 수치, 추후 쌓이는 데이터에 따라 변경 될 수 있음)

3. 프론트 엔드 - 격상 제안 알림

웹소켓을 통해 백 엔드 서버로부터 재난 대응 단계 격상에 대한 정보를 수신하고 정보를 조회용 페이지에 알림으로 보여준다.

6-3. 필요한 기술 요소 및 개발

1. 프론트 엔드

1. HTML, CSS, JS를 이용한 서버와 통신을 할 수 있는 간단한 웹페이지 제작

2. 백 엔드

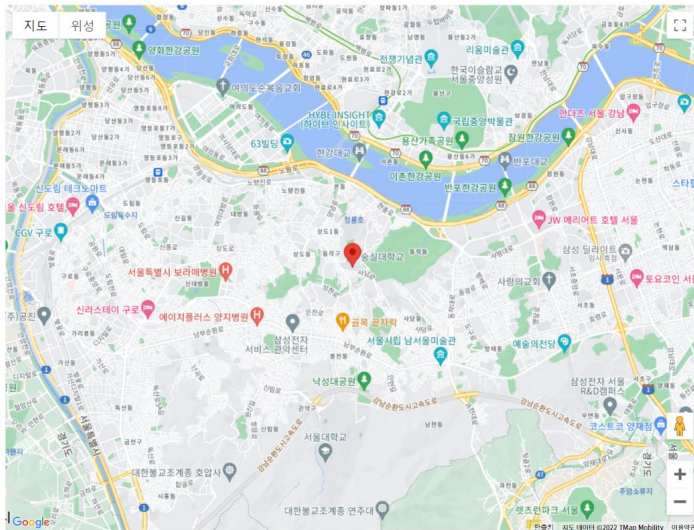
1. 서버: Node.js 기반 fastify 사용
2. DB: sqlite3 사용

* 격상 여부 판별: 초기에는 rule-based AI (알고리즘) 기반, 프로젝트의 확장 가능성에 따라 Deep learning based AI 기반으로 교체할 수 있음.

6-4. 구현

Git repository: https://github.com/XQtbl/opensrc_j4
(위의 링크에서 프로젝트의 모든 코드를 확인할 수 있음)

서비스 이용해 볼 수 있는 주소: <https://3770.xqtbl.net/?>



[4] 구현 - 신고 제출 페이지

6-5. 개선방안

현재 프로젝트는 신고에 대해 직접 입력을 해 주어야 시스템에서 확인하고 처리를 할 수 있다. 이 부분에 있어서 통화 내역을 분석해 자동으로 신고 포맷에 맞게 자동으로 입력 및 서버로 전송해주는 AI 를 도입한다면 더 나은 사용성을 가질 수 있을 것으로 보인다. 또한 긴급신고 통합시스템에서 다른 기관으로 들어온 신고를 본 서비스에 적용하는 것은 현재 프로젝트에서는 무리가 있다고 보인다. 현재의 통합시스템에서 처리를 할 때 본 프로젝트의 서버로도 함께 전송하는 방식을 사용한다면 이 문제는 해결될 수 있다.

7. 결론

기존 긴급 신고 대응 시스템은 재난 대응 단계 격상을 사람이 판단 및 현재 신고 상황을 항시 모니터링 하고 있다. 본 프로젝트에서는 신고를 접수하면 신고의 세부 내용과 위치 정보, 심각도(기존의 재난 대응 단계)를 데이터베이스에 저장한다. 또한 서버에서는 신고 내역 데이터 베이스를 바탕으로 신고의 접수 빈도, 위치 (위도 및 경도), 대응 단계를 파악한 후 단계 조정에 관한 알림을 신고 조회용 페이지로 전송한다. 이를 통해 직원은 신고 업무에만 집중할 수 있게 되고, 혹시 사람이 놓칠 수 있는 부분에 대해서 본 프로젝트가 도움을 줄 수 있다. 즉, 본 프로젝트를 통해 반복되고 유사한 신고를 소방서 상황실 직원들에게 상기시킴으로 재난에 빠른 대응이 가능하고 프로그램으로 신고현황에 대한 모니터링을 자동화하여 사고를 줄일 수 있다. 현재의 시스템에는 없는 본 프로젝트의 아이디어와 제작한 시스템을 새로이 도입함으로 소방서 혹은 외의 긴급 신고를 받는 기관에서도 분명 얻을 수 있는 이점이 많을 것으로 예상된다.

8. 참고문헌

행정안전부, *Ministry of the Interior and Safety.*, 긴급신고 통합서비스, (2022)

<https://www.mois.go.kr/frt/sub/a06/b10/emergencycall/screen.do>

소방, 경찰 ‘공조 요청’ 2 번 무시… 신고자엔 “112 연락하라”, 2022, 이홍근, 권정혁,

<https://www.khan.co.kr/national/incident/article/202211082111005>

소방, 경찰에 2 시간 동안 15 차례 지원 요청 [이태원참사 8 시간의 기록], 곽진산, 2022,

https://www.hani.co.kr/arti/society/society_general/1068696.html

'이태원 참사' 112 신고 녹취록 파장... "참사 방치", YTN, 2022

https://www.ytn.co.kr/_ln/0103_202211020645330111

Everyday Types, TypeScript, *Microsoft 2012-2022*,

<https://www.typescriptlang.org/docs/handbook/2/everyday-types.html>

Date.prototype.toLocaleString() document, 2022, *mdn web docs*,

https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Date/toLocaleString

fastify Documentation, *2016-2022 OpenJS Foundation and The Fastify team*,

<https://www.fastify.io/docs/latest/>

TryGhost/node-sqlite3 API wiki, 2022, [https://github.com/TryGhost/node-](https://github.com/TryGhost/node-sqlite3/wiki/API)

[sqlite3/wiki/API](https://github.com/TryGhost/node-sqlite3/wiki/API)

NodeJS - 거리구하기(위도 / 경도), neity, 2020,

<https://velog.io/@neity16/NodeJS-%EA%B1%B0%EB%A6%AC%EA%B5%AC%ED%95%98%EA%B8%B0%EC%9C%84%EB%8F%84-%EA%B2%BD%EB%8F%84>