**캡스톤 디자인 I**

**종합설계 프로젝트**

|  |  |
| --- | --- |
| 프로젝트 명 | *캡스톤 계획서 안내* |
| 팀 명 | *캡스톤 팀* |
| 문서 제목 | 계획서 |

|  |  |
| --- | --- |
| **Version** | 1.2 |
| **Date** | 29 |

|  |  |
| --- | --- |
| **팀원** | 홍 길동 (조장) |
| 이 순신 |
| 왕 건 |
| 연개소문 |
|  |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “xxxx xxxx”를 수행하는 팀 “xxxxx”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “xxxxxx”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | 계획서-유하.doc |
| **원안작성자** | 이태훈 |
| **수정작업자** | 홍길동, 이순신, 왕건 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2020-03-09 | 이태훈 | 1.0 | 최초 작성 |  |
| 2020-03-11 | 이인평 | 1..1 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**목 차**

[**1** **개요** 4](#_Toc347412182)

[1.1 프로젝트 개요 4](#_Toc347412183)

[1.2 추진 배경 및 필요성 4](#_Toc347412184)

[**2** **개발 목표 및 내용** 5](#_Toc347412185)

[2.1 목표 5](#_Toc347412186)

[2.2 연구/개발 내용 6](#_Toc347412187)

[2.3 개발 결과 7](#_Toc347412188)

[2.3.1 결과물 목록 및 상세 사양 7](#_Toc347412189)

[2.3.2 시스템 기능 및 구조 7](#_Toc347412190)

[2.4 기대효과 및 활용방안 7](#_Toc347412191)

[**3** **배경 기술** 8](#_Toc347412192)

[3.1 기술적 요구사항 8](#_Toc347412193)

[3.2 현실적 제한 요소 및 그 해결 방안 9](#_Toc347412194)

[3.2.1 하드웨어 9](#_Toc347412195)

[3.2.2 소프트웨어 9](#_Toc347412196)

[3.2.3 기타 9](#_Toc347412197)

[**4** **프로젝트 팀 구성 및 역할 분담** 10](#_Toc347412198)

[**5** **프로젝트 비용** 10](#_Toc347412199)

[**6** **개발 일정 및 자원 관리** 11](#_Toc347412200)

[6.1 개발 일정 11](#_Toc347412201)

[6.2 일정별 주요 산출물 12](#_Toc347412202)

[6.3 인력자원 투입계획 13](#_Toc347412203)

[6.4 비 인적자원 투입계획 14](#_Toc347412204)

[**7** **참고 문헌** 15](#_Toc347412205)

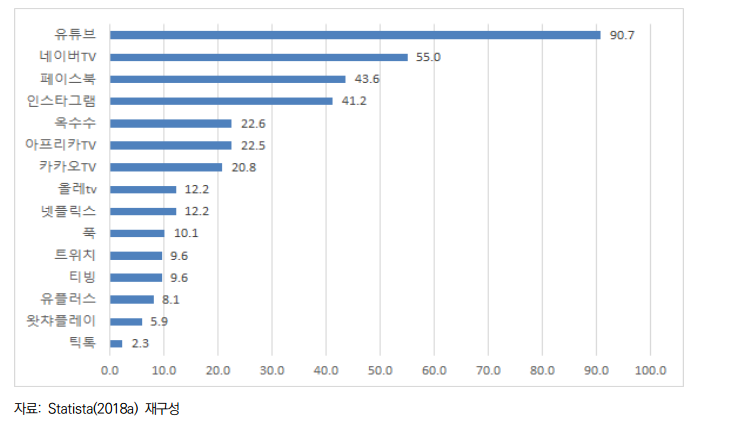
# **개요**

## 프로젝트 개요

우리 프로젝트는 동영상에서 영상과 음성을 분리해 각각 사전 학습된 딥러닝 모델을 이용해 라벨링을 실시하고, 만약 미성년자에게 부적합한 영상 및 음성이 발견된다면 해당 구간을 이용자에게 알려주고, 어떤 라벨에 의해 발견되었는지 알려준다.

## 추진 배경 및 필요성

우리는 현재 스마트폰 및 태플릿 PC 등 다양한 환경에서 쉽게 인터넷에 접속하고, 많은 영상을 시청한다. 예를 들면 대표적인 동영상 업로드 사이트인 유튜브가 있는데, 닐슨 코리아의 통계에 따르면 2019년 8월 기준 유튜브 이용자 수가 2,631만명에 달한다. 이는 한국에서 서비스를 하고 있는 영상 플랫폼(OTT 동영상 플랫폼)들 중 가장 높은 숫자를 기록한다.



**<사진 1. 동영상 플랫폼 이용자 현황>**

그런데, 유튜브 영상 업로드 시스템은 매우 이용하기 편리하고 접근성이 좋다는 장점이 있지만, 누구나 업로드가 가능하다는 시스템이 미성년자를 포함하여 성인들에게도 부적합하거나 부정적인 영향을 끼치는 영상이 쉽게 발견되고는 한다. 현재 유튜브에는 다양한 가이드 라인이 존재하며, 해당 가이드 라인에 적합하지 못하는 영상들은 업로드하지 못하게 되어있다.



**<사진 2. 현재 Youtube 영상 검열 시스템>**

하지만 현재 유튜브에서는 이러한 영상을 실시간 모니터링과 시청자들의 신고로 파악하는 것이 현재 실태이다. 따라서 어떤 영상들은 미성년자에게 부적합한 영상임에도 미성년자에 대한 제약을 걸지 않았고, 조회수도 상당히 높은 것을 발견할 수 있다. "*신세계*"라는 영화로 예를 들자면, "*신세계*"라는 영화는 청소년 관람 불가 영화이다 Youtube에서도 또한 확인할 수 있다.



**<사진 3. 신세계 영화 - Youtube 제공>**

그러나 해당 영화에서 선정적이거나 폭력적인 장면이 고스란히 담겨있는 장면들을 짜집기 해 놓은 영상들은 누구나 시청 가능하게 설정되어 있고, 어떠한 영상도 Youtube 정책에 의해 검열되지 않았다. 심지어 400만이 넘는 조회수를 기록한 영상도 존재했다.



<사진 4. 신세계 명장면 - Youtube>

미성년자에게 부적합한 영상은 다음과 같이 개제되어야한다.

`

<사진 5. 미성년자 부적합 영상 - Youtube>

그러나 사진4과 같은 예로 많은 영상들이 미성년자에게 부적합한 내용을 담고 있음에도 불구하고, 사진 5와 같은 화면이 나오지 않기 때문에 미성년자들은 성인 인증과 같은 절차를 거치지 않아도 영상을 볼 수 있다. 아래 사진 6의 내용은 여성가족부가 실시한 청소년 매체 이용 및 유해 환경 실태에 대한 설문조사 결과이다. 내용에 따르면, 조사에 응한 청소년 중 반이 넘는 인원들이 거의 매일 Youtube를 사용한다고 응답했고 41% 에 해당하는 청소년들은 청소년 유해 영상에 접근했을 때 성인 인증 절차를 밟지 않고 접근할 수 있었다고 한다. 이를 통해 우리는 많은 청소년들이 Youtube를 통해서 접해서는 안될 많은 청소년 부적합 영상들을 시청하고 있다는 것을 알 수 있다. 따라서 우리는 미성년자들에 초점을 맞춰서 해당 연령대에 부적합한 영상에 대해 19세 이상 시청가능 조건을 걸지 않으면 Youtube에 업로드 하지 못하도록 하는 자동 필터링 시스템을 개발하기로 했다.

****

**<사진 6. 청소년들의 Youtube 이용 실태>**

# **개발 목표 및 내용**

## 목표

- 한국에서 가장 많이 이용되고 있는 영상 플랫폼인 Youtube의 가이드라인에 맞게 동영상을 업로드 할 수 있고, 시청할 수 있게 하여 건전하고 건강한 Youtube를 이용할 수 있게 한다.

- Youtube에 업로드 하기 전 필터링 과정을 거치며 각 영상과 음성에 대한 필터링이 실시되어 Youtube 가이드라인에 부적합한 구간이 발견되면 업로드하고자 하는 이용자에게 해당 구간을 알려주고, 어떤 가이드라인에 의해 부적합한지 알려준다.

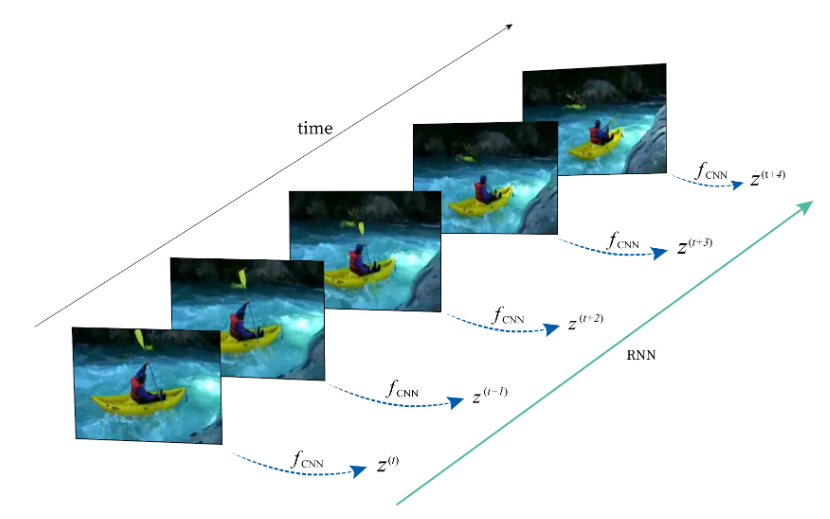
- 현재 Youtube에 존재하는 많은 가이드라인 중 우리는 선정적인 것, 폭력적인 것, 담배, 욕설에 대한 가이드라인을 충족시키는 지 확인하는 필터링을 실시할 것이다.

## 연구/개발 내용

**1. 영상 필터링**

딥러닝 모델을 이용하여 영상 필터링을 진행한다. 필터링 되는 내용은 폭력적인 장면(만 19세 이상 영화에서 추출한 흉기에 찔리거나 베이는 장면, 만 18세 이상 게임에서 추출한 총, 흉기에 의해 피가 튀는 장면), 선정적인 장면(여성의 과도한 노출), 담배에 해당된다. 영상 필터링 시스템의 세부 모델은 다음과 같다.

- 해당 데이터셋으로 학습된 딥러닝 모델을 준비한다. 딥러닝 모델은 CRNN을 이용해 설계되었으며, ResNetCRNN을 채택하여 학습을 진행하였다. 학습 데이터는 Youtube를 이용해 직접 구축하였다.ResNetCRNN의 작동원리는 다음과 같다.



- 업로드된 영상의 길이를 파악한다. 30분 이하 길이를 가지는 영상은 모든 구간(10초)을 학습된 딥러닝 모델로 Test해서 부적합한 구간인지 판단한다. 30분을 초과하는 영상에 대해서는 총 Sampling한 길이가 30분이 되도록 10초씩 Random Sampling하여 딥러닝 모델로 Test해서 부적합한 구간인지 판단한다.

**2. 음성 필터링**

딥러닝 모델을 이용하여 음성 필터링을 진행한다. 필터링 되는 내용은 한국어로 표현된 욕설에 해당되며, 다양한 비속어를 포함한다.

- STT(Speech To Text)기술을 이용하여 음성을 텍스트로 변환한다. STT는 현재 구글 API를 이용해서 진행하고 있으며, 만약 정확도가 낮다고 판단될 시 Kaldi 기반의 한국어 음성 인식기인 Zeroth를 사용하여 딥러닝을 통한 STT를 구현한다.

- 카카오에서 개발한 khaiii(Kakao Hangul Analyzer III) 형태소 분석기 모델을 통해 음성에서 추출된 텍스트를 형태소 단위로 전 처리한다. khaiii는 CNN 알고리즘을 이용하여 형태소 분석을 구현하고, 신경망이 오 분류할 경우를 대비하여 알고리즘 앞단과 뒷단에 사용자 사전 장치를 마련하여 사용자가 설정한 입력 어절은 설정한 값 자체로 분석하도록 한다. 이러한 형태소 분석 전처리를 함으로써 텍스트에 존재하는 욕설들은 모두 형태소로 처리되고, 이로써 전처리된 텍스트는 음성 필터링에 있어 매우 적합한 데이터가 된다.

- FastText 모델을 이용하여 텍스트안에 존재하는 형태소와..... FastText는 Word2Vec을 기본으로 하지만 부분 단어들을 임베딩하는 기법으로, 주변에 있는 단어들을 가지고 중심에 있는 단어를 맞추는 cbow 모델 또는 중심에 있는 단어로 주변 단어를 예측하는 skip-gram 모델로 구현될 수 있다.

**3. AWS 및 웹서버**

동영상이 업로드 되고, 필터링의 결과가 반영되는 웹 페이지를 만들고 웹 페이지의 배포를 위해 웹서버를 구축한다. 또한 딥러닝 모델 학습과 모델의 영상과 음성 필터링에 있어 필요한 Amazon EC2 instance와 Database를 생성한다.

- 사용자 인터페이스를 만들기 위한 JavaScript 라이브러리인 React를 이용하여 웹페이지를 제작한다. 웹 구현을 위해 이용되는 React의 component 중 우리 프로젝트에 있어 많은 비중을 차지하는 것은 동영상 업로드 box, 영상 필터링 결과 box, 음성 필터링 결과 box이다. component마다 필요한 기능에 따라 적합한 라이브러리를 채택하고 CSS를 적용하여 레이아웃과 스타일을 정의한다.

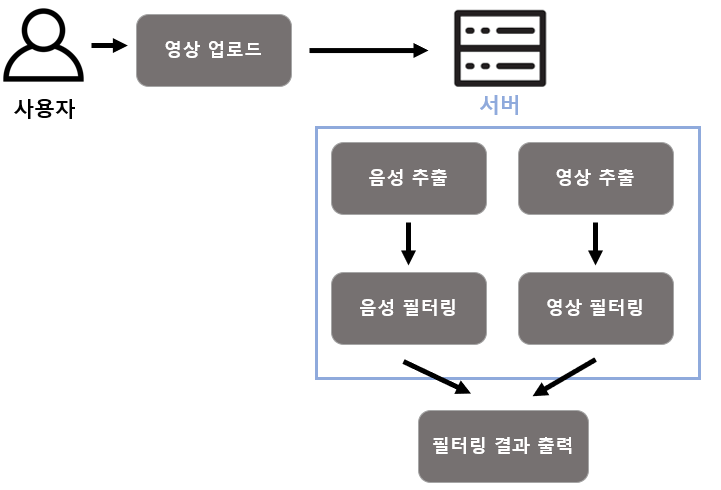
- React로 구현한 웹 페이지 배포를 위해 apache server를 웹 서버로 선택한다.

- AWS EC2 instance는 AWS Deep Learning AMI를 채택하여 딥러닝 모델의 학습을 진행한다. 모델 학습에 있어 필요한 data set은 Amazon Elastic Block Storage(EBS)에 저장한다.

- 태깅이 완료된 AWS EC2 인스턴스의 Apache server로 배포된 웹 페이지 그 결과를 반영하여

## 개발 결과

### 시스템 기능 요구사항



### 시스템 비기능(품질) 요구사항

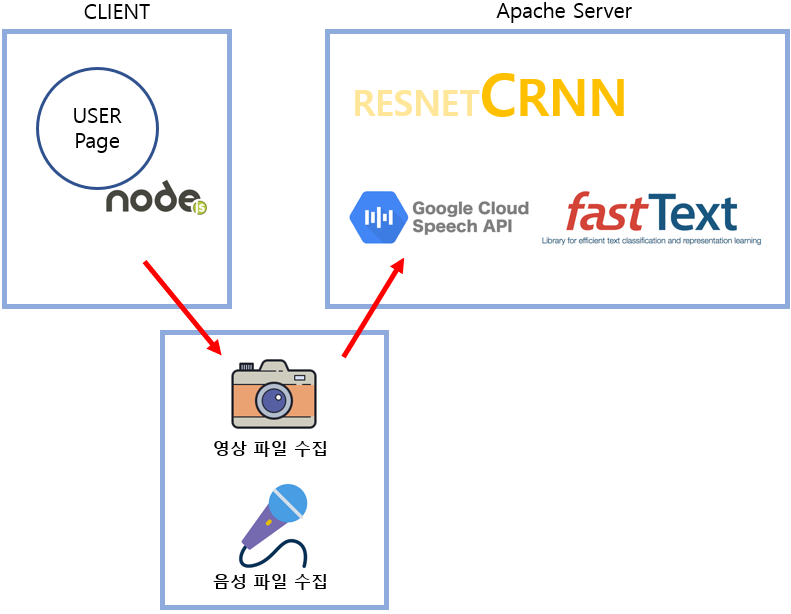
**사용성 : 사용자가 한 눈에 사용방법을 알 수 있도록 UI를 제작한다**

**필터링 결과를 한 눈에 알아볼 수 있도록 가시화한다**

**기능성 : 보안성을 위해 필터링이 끝난 데이터는 즉시 데이터베이스에서 삭제한다**

**효율성 : 필터링에 소요되는 시간은 최대 10분이 넘어가지 않도록 설게한다**

### 시스템 구조



### 결과물 목록 및 상세 사양

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 대분류 | 소분류 | 기능 | 형식 | 비고 |
|  | *프레임 분할* | 음성 프레임을 분할한다 | 함수 |  |
|  | *음성 추출* | 영상 파일에서 음성을 추출한다 |  |  |
|  | *STT* | 음성을 텍스트로 변환한다 |  |  |
|  | *영상 필터링* |  |  |  |
|  | *음성 필터링* |  |  |  |
|  | *영상 업로드* |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 기대효과 및 활용방안

먼저 우리의 필터링 시스템은 Youtube의 가이드 라인에 맞춰서 영상과 음성을 검열한다. 이는 현재 신고 기반으로 이루어지는 수작업 검열 과정보다 훨씬 효율적으로 작동되는 업로드 이전에 일어나는 자동으로 검열하는 시스템이므로 다양한 기대효과와 활용방안이 있다.

1. 필터링 시스템을 적용함으로써 청소년들에 있어서 부적합한 영상들을 1차적으로 검열할 수 있어서, 동영상 업로드 플랫폼에서 실시하는 신고 기반 검열 시스템보다 더 많은 영상을 검열할 수 있고, 작업량과 비용 측면에서 효율적이다.

2. 우리의 시스템은 Youtube 뿐만 아니라, 현재 많은 이용자들이 이용하는 실시간 스트리밍 서비스(Twitch, Affreca TV 등)에서도 효과적으로 사용할 수 있다. 각 스트리밍 서비스들은 운영진이 직접 실시간 모니터링과 시청자들의 신고를 통해 제재가 가해지는데, 그 숫자가 매우 많아 효과적으로 이루어지지 않는다. 따라서 방송되고 있는 장면들을 실시간으로 딥러닝 모델에 넣어 검열할 수 있다.

3. 이러한 검열 작업이 원활하게 이루어지면, 현재 미성년자들이 무분별하게 시청하고 있는 영상들이 자동으로 검열됨으로써 청소년에게 유해한 매체물과 약물 등이 청소년에게 유통되는 것과 청소년이 유해한 업소에 출입하는 것 등을 규제하고 청소년을 유해한 환경으로부터 보호ㆍ구제함으로써 청소년이 건전한 인격체로 성장할 수 있도록 함을 목적으로 한다.

4. 현재는 미성년자들을 대상으로 부적합한 영상들을 검열하는 시스템을 만들었지만, 점차 데이터셋이 많아지고 다양한 라벨에 대해서 학습을 시킨다면, 교통사고, 자연 재해, 전쟁 또는 길거리 싸움, 집단 구타 및 학대 등 검열 대상을 확대시킬 수 있다.

5. 또한 음성 필터링은 단순 욕설에 대해서만 진행되지만, 데이터셋 양을 확대시켜 학습시킨다면 본질적 속성을 토대로 한 악의적인 모욕(인종차별적 비방), 미성년자에게 수치심을 주거나 모욕감을 주는 등 검열 대상을 확대시킬 수 있다. 여기서 본질적 속성이란, 보호 대상 집단 신분, 신체적 특징 또는 성폭행, 가정폭력, 아동 학대 등의 피해자 신분이다.

6. 우리의 시스템이 적극적으로 활용된다면, 해당 동영상 업로드 플랫폼에 대한 사람들의 신뢰도와 인식의 향상에 도움이 된다. 또한 미성년자 뿐만 아니라 일반인에 대해서도 각종 범죄에 쉽게 노출되지 않게 함으로써, 많은 사회적 범죄를 예방할 수 있다.

# **배경 기술**

## 기술적 요구사항

**1) 프로젝트 개발**

1. 웹페이지 개발 언어

- Javascript(react 포함), Css

2. 웹페이지 개발 환경

- Window 10, Visual Studio Code, Chrome

3. 서버 개발 언어

- Javascript(소켓 통신), Php(파일 업로드)

4. 서버 개발 환경

- AWS EC2 인스턴스(Apache 웹서버), AWS Cloudfront, Linux(Ubuntu 16.04),

Apache Web Server

5. 영상, 음성 필터링 개발 언어

- Python

6. 영상, 음성 필터링 라이브러리

- Pytorch(Video Classification 딥러닝 모델), Kaldi - Zeroth(STT 딥러닝 모델),

FastText(욕설 판별 딥러닝 모델)

7. 영상, 음성 필터링 개발 환경

- AWS EC2 인스턴스(Deep Learning AMI), Linux(Ubuntu 16.04), Jupyter Notebook

**2) 프로젝트 결과**

1. 웹페이지(Chrome)

- 동영상을 업로드하고, 필터링된 출력 결과를 확인한다. 업로드 파일의 확장자

는 .avi, .mp4로 제한하며, 그에 대한 주의사항이 표기되어있는 UI를 제공한다.

## 현실적 제한 요소 및 그 해결 방안

**3.2.1 하드웨어**

1. 모든 딥러닝 모델(Video Classification, Kaldi - Zeroth, FastText)을 학습시키기 위해서는 일반적인 로컬(데스크탑 PC, 노트북 등)에서는 RAM 메모리와 GPU 메모리의 양이 절대적으로 부족해 원활하게 진행되지 않는다. 따라서 우리는 AWS의 Deep Learning AMI를 이용해 학습 및 테스트를 진행한다.

2. Internet Explorer에서는 우리가 만든 웹페이지 기능이 정상적으로 작동되지 않을 수 있다. 각종 라이브러리가 제대로 적용되지 않기 때문이다. 이를 크리에이터들이 Chrome에서만 접속하여 이용할 수 있도록, 주의사항을 작성해 해결한다.

**3.2.2 소프트웨어**

1. 단순 AWS EC2 인스턴스 도메인으로 대용량 FIle Upload를 진행할 시 속도가 느려 AWS CloudFront에서 인스턴스를 생성해 기존 EC2 인스턴스와 연결하여 도메인을 새로 설정해 Upload 속도를 빠르게 하여 해결할 수 있다.

2. 음성 필터링을 진행할 때 Google API만 사용해서는 Youtube 환경에서 음성을 텍스트로 바꾸는 STT기술이 정확하게 작동하지 않는다. 따라서 Kaldi - Zeroth 딥러닝 모델을 통해 직접 학습시킴으로써 해결한다.

3. '빨간 원숭이의 해'(예: 2016년)를 뜻하는 '병신년'과 시작점을 뜻하는 '시발점'등 욕설이 아님에도 불구하고, 욕설과 같은 글자들을 오분류하는 문제점이 발생한다. 따라서 FastText 딥러닝 모델을 이용하여, 문맥을 고려함으로써 해결한다.

**3.2.3 기타**

1. 모든 Youtube 가이드라인을 맞추어서 필터링을 진행하기에는 그 데이터셋 구축하는 시간과 노력이 상당하고, 단순 Video Classfication으로 필터링할 수 없어서, 가이드라인 중 4개에 충족되게끔 목표를 설정했다.

# **프로젝트 팀 구성 및 역할 분담**

**프로젝트에 참여하는 멤버의 역할을 구체적으로 명시한다.**

| 이름 | 역할 |
| --- | --- |
| 홍길동 | * Software Project Leader |
| 이순신 | * User Interface 담당 * Web Interface 개발 |
| 왕건 | * DB 설계 및 DB Query 시스템 개발 |
|  |  |

# **프로젝트 비용**

**프로젝트에 투입될 비용을 예상하여 기입한다. 여기서 Man-Days 라 함은 인적 비용을 계산하기 위한 단위로 한 사람이 하루 8시간 정도 일을 해야 하는 양을 1 MD라고 한다. 즉, 한 사람이 하루 4시간씩 일을 하게 되면 이틀 정도 일을 했을 때 1 MD 정도의 비용이 들어갔다고 한다.**

|  |  |
| --- | --- |
| **항목** | **예상치 (MD)** |
| 커널 | 30 |
| 미들웨어 | 20 |
| 개발 환경 | 60 |
|  |  |
|  |  |
|  |  |
|  |  |
| 합 |  |

# **개발 일정 및 자원 관리**

## 개발 일정

**개발 일정을 계획한다.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **항목** | **세부내용** | **1월** | **2월** | **3월** | **4월** | **5월** | **6월** | **비고** |
| 요구사항분석 | 요구 분석 |  |  |  |  |  |  |  |
| SRS 작성 |  |  |  |  |  |  |  |
| 관련분야연구 | 주요 기술 연구 |  |  |  |  |  |  |  |
| 관련 시스템 분석 |  |  |  |  |  |  |  |
| 설계 | 시스템 설계 |  |  |  |  |  |  |  |
| 구현 | 코딩 및 모듈 테스트 |  |  |  |  |  |  |  |
| 테스트 | 시스템 테스트 |  |  |  |  |  |  |  |

## 일정별 주요 산출물

**일정별로 어떤 결과물을 도출할 지 상세하게 작성한다. 그래프의 형태로 작성하여도 좋다.**

|  |  |  |  |
| --- | --- | --- | --- |
| 마일스톤 | 개요 | 시작일 | 종료일 |
| 계획서 발표 | 개발 환경 완성 (GCC 설치, 기본 응용 작성 및 테스트 완료)  **산출물 :**   1. 프로젝트 수행 계획서 2. 프로젝트 기능 일람표 | ~ | 2012-03-08 |
| 설계 완료 | 시스템 설계 완료  **산출물 :**   1. 시스템 설계 사양서 | 2012-03-09 | 2012-03-20 |
| 1차 중간 보고 | 기능 xxx ~ yyy 구현 완료  **산출물 :**   1. 프로젝트 1차 중간 보고서 2. 프로젝트 진도 점검표 3. 1차분 구현 소스 코드 | 2012-03-21 |  |
| 2차 중간 보고 | 기능 zzz ~ xyz 구현 완료  **산출물 :**   1. 프로젝트 2차 중간 보고서 2. … |  |  |
| 구현 완료 | 시스템 구현 완료  **산출물:** |  |  |
| 테스트 | 시스템 통합 테스트  **산출물:** |  |  |
| 최종 보고서 | 최종 보고  **산출물:** |  |  |

## 인력자원 투입계획

**프로젝트 참여 인력이 언제부터 언제까지 어떤 일로 투입이 될 지 구체적으로 명시한다.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 이름 | 개발항목 | 시작일 | 종료일 | 총개발일(MD) |
| 홍길동 | *디바이스 드라이버* | 2009-03-01 | 2009-04-20 | 20 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 비 인적자원 투입계획

**개발 환경 등 비 인적 자원의 투입 계획을 명시한다.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 항목 | Provider | 시작일 | 종료일 | Required Options |
| 컴파일러 | Microsoft | 2009-03-01 | 2009-04-20 |  |
| 개발용 PC 4대 | Dell |  |  |  |
| 임베디드 보드 | 미정 |  |  | PXA270 |
|  |  |  |  |  |

# **참고 문헌**

**참고한 서적, 기사, 기술 문서, 웹페이지를 나열한다.:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 번호 | 종류 | 제목 | 출처 | 발행년도 | 저자 | 기타 |
|  | 서적 |  |  |  |  |  |
|  | 기사 |  |  |  |  |  |