**캡스톤 디자인 I**

**종합설계 프로젝트**

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

|  |  |
| --- | --- |
| **Version** | 1.0 |
| **Date** | 04 |

|  |  |
| --- | --- |
| **팀원** | 전호현(조장) |
| 이근하 |
| 유문상 |
| 최원대 |
|  |

|  |
| --- |
| **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-프로젝트명.doc |
| **원안작성자** | 전호현, 이근하, 유문상, 최원대 |
| **수정작업자** | 전호현, 이근하, 유문상, 최원대 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2018-03-05 | 전호현 | 1.0 | 최초 작성 | 목차 별 초안 작성 |
| 2018-03-06 | 최원대 | 1.1 | 내용 추가 | 목차 3,5,6 추가 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**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

규모가 있는 SW 개발을 함에 있어 동일한 프로젝트에 대해 여러 개발자가 동시에 개발할 수 있도록 돕는 버전 관리 시스템을 사용하는 것은 필수적이다. 따라서, 개발자라면 버전 관리 시스템을 효율적으로 사용할 수 있어야 한다.

그렇다면, 버전 관리 시스템을 사용하여 개발한 프로젝트에서 누가, 얼마나 개발에 기여했는지를 간략하게 보여줄 수는 없을까? 우리 팀은 버전 관리 시스템 중에서도 가장 많이 쓰이는 GitHub를 활용한 프로젝트의 기여도 평가를 돕는 웹 서비스를 만들고자 한다. 프로젝트 참여자 각각의 커밋 횟수, 삽입 라인, 삭제 라인, 코드의 안정성 등에 대한 정보를 제공하여, 각자의 기여도를 평가해 볼 수 있다.

## 추진 배경 및 필요성

### 추진 배경

### Git Inspector 의 장점

### Git Inspector의 문제점

### (새로 개발할 프로젝트명)의 필요성

**본 프로젝트를 수행하게 된 배경과 사전 조사 내용을 서술하며, 프로젝트에서 개발할 시스템의 필요성에 대하여 명확하게 기술한다.**

**기존의 시스템을 보완하는 경우에는 논리적으로 귀납법적인 논리를 전개한다. 즉, 현재 기 운용되고 있는 시스템은 이러저러한 문제점이 있다고 설명하고 이러한 문제점을 해결하기 위한 시스템 개발이 필요하다는 식으로 서술한다. 또는 기존의 시스템에서 개선되면 더 좋은 시스템이 될 가능성이 있는 기능들이 있음을 기술한다. 이 부분은 매우 설득력이 있게 기술하여야 한다. 더불어, 이러한 시스템의 시장환경, 발전환경 등의 부가적인 설명도 기술한다.**

**(예: 1.2.1 ...기술의 시장 현황, 1.2.2 ... 기술 발전 현황,**

**1.2.3 ... 기 개발된 시스템 현황...**

**1.2.4 기 개발된 시스템의 문제점 혹은 개발할 시스템의 필요성)**

# **개발 목표 및 내용**

## 목표

본 프로젝트의 목표는 기존의 Git Inspector를 보완하여, GitHub를 활용해 프로젝트를 진행한 참여자들 각각의 기여도를 평가할 수 있도록 돕는다. 명령어 Command를 사용해야 하는 Git Inspector를 UI를 제공하여 편리하게 사용할 수 있으며, 데이터베이스에 정보를 미리 저장하여 더 빠른 분석결과를 제공한다.

이를 통해 기업에서 각 개개인의 기여도를 평가하거나, 대학에서 프로젝트 과목의 학점을 평가하는 데 도움이 될 수 있을 것이다.

**프로젝트의 목표를 명확하게 제시한다.**

**예시) ooo를 위한 ooo가 되는 ooo를 개발 한다.**

## 연구/개발 내용

### GitHub 프로젝트의 변동사항을 알려줄 API 탐색

### AWS Lambda

### 데이터베이스에 분석결과 데이터 저장

### ServerSide 구축

### Client Side 구축

**본 프로젝트의 수행의 내용을 구체적으로 기술한다.**

**목표를 세분화하여 세부 목표를 정하고 그에 따른 결과물을 제시한다.**

**연구/개발 방법을 기술한다. 연구/개발 방법은 단계별 수행 방법을 기술한다.**

## 개발 결과

### 시스템 기능 요구사항

**USE-CASE DIAGRAM**

**기능 요구사항은 시스템의 특성에 따라 적합한 형태로 서술하되, 가급적 유즈케이스 다이어그램을 사용한다.**

### 시스템 비기능(품질) 요구사항

**비기능(품질) 요구사항 서술 시, “이 시스템의 성능은 동시 접속자 수 1000명일 때, 초당 10000 트랜잭션을 처리할 수 있어야 한다. “ 와 같이 구체적으로 명시한다.**

**비기능(품질) 요구사항이 2가지 이상일 경우, 요구사항의 우선수위를 함께 명시한다. 예를 들어, 시스템이 만족해야 하는 비기능 요구사항이 성능과 보안이라면, 두 요소가 모두 만족되지 못할 경우, 보안을 위해 성능을 포기할 수 있다면 보안이 성능보다 우선순위가 높아야 한다.**

### 시스템 구조

**시스템의 전체적인 구조를 파악할 수 있는 구조를 도식화하는 아키텍처를 제시하고 아키텍처의 각 구성요소를 설명한다. 시스템이 외부 시스템과 연동된다면, 외부 시스템까지 포함하여 도식화한다. 아키텍처는 지속적으로 변경될 수 있으나, 현재 계획서에서 포함하고 있는 기능 및 비기능적 요구사항은 모두 반영된 구조를 제시하여야 한다.**

**참고 문서**

* <http://capstone.cs.kookmin.ac.kr/gongjisahang-1/swgonghagteuggang>

### 결과물 목록 및 상세 사양

**프로젝트 수행의 결과물을 목록으로 제시하고 이에 대한 상세 사양을 기술한다. 상세 사양은 결과물에서 제공하는 기능들을 프로젝트 수행의 진도를 평가할 수 있는 기능 일람표를 아래 예제와 같이 작성한다. 작성된 기능 일람표는 향후, 테스트 케이스의 기준으로 활용할 수 있도록 상세하게 작성한다.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 대분류 | 소분류 | 기능 | 형식 | 비고 |
| 파일 | *파일 저장* | 현재 열린 파일을 저장한다. | DLL/함수 |  |
|  |  | 다른 이름으로 파일을 저장한다 | DLL/함수 |  |
|  | *파일 열기* | 문서 파일을 연다. | 모듈 |  |
| 출력 | *PDF* |  |  |  |
|  | *프린터* |  |  |  |
| 모양 | *글꼴* | 정렬 |  |  |
|  |  | 폰트 바꾸기 |  |  |
|  |  |  |  |  |

## 기대효과 및 활용방안

# **배경 기술**

## 기술적 요구사항

* Back-end
* Github Webhook : Github의 레포지터리에 특정 이벤트가 발생하는지를 탐지하는 API. Webhook을 이용해서 이벤트가 발생하면 HTTP Post 페이로드를 AWS API Gateway로 보낸다.
* AWS API Gateway : 개발자가 API를 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있게 해주는 AWS 서비스. Github Webhook으로 받은 데이터를 AWS Lambda로 보낸다.
* AWS Lambda : 이벤트에 응답하여 코드를 실행하는 서버 없는 컴퓨팅 서비스이다. 본 프로젝트에서는 두 번의 AWS Lambda를 이용하는데, 첫 번째는 AWS API Gateway에서 받은 Github 레포지터리의 정보를 clone 해서 Amazon S3에 저장하기 위해 사용한다. Git inspector를 사용할 때 항상 레포지터리를 clone하기 때문에 다소 시간이 걸린다. 이를 Amazon S3에 미리 clone 시켜 정보를 불러오는데 시간이 줄이기 위함이다. 두 번째 AWS Lambda는 사용자가 웹 어플리케이션 상에서 정보를 요구하면 Git inspector를 통해서 Amazon S3에 저장되어 있는 레포지터리의 정보를 pull 해서 결과를 보여주기 위해 사용된다. Java, Node.js , C# 및 Python을 지원한다.
* Amazon S3 : 아마존에서 제공하는 객체 스토리지로, 원하는 양의 데이터를 저장하고 검색할 수 있도록 구축 되어있다. 위에서도 말했듯이 S3에는 Github 레포지터리의 clone을 저장하기 위해 사용한다.
* Git inspector : Github 레포지터리를 분석해주는 오픈소스이다. 레포지터리의 주소를 입력하면 commit 횟수, 삽입한 line 수 , 삭제한 line 수 등등에 대한 정보를 나타내준다.
* Front-end
* 웹 어플리케이션 : Javascript, HTML, CSS 그리고 Ajax 를 이용해서 유저가 사용하는 웹 어플리케이션을 제공한다. 유저가 Github의 주소와 조건을 입력하면 웹을 통해서 해당 주소의 레포지터리를 분석해서 조건에 맞는 정보를 제공한다.

## 현실적 제한 요소 및 그 해결 방안

### 하드웨어

### 소프트웨어

1. Git inspector가 제공하는 정보만으로는 Git을 잘 사용하고 있는지를 알 수 없을 수도 있다. Git inspector는 현재 commit 횟수, 삽입 line, 삭제 line, 변화 비율, 유저가 적은 모든 줄의 평균 지난 일 수(age) ,유저가 적은 라인이 지금까지 살아남아 있는 비율(stability) 그리고 전체 줄 수 대비 주석 비율을 보여주고 있다. 이는 Git에 대해 잘 알고 있다면 이러한 정보를 통해 레포지터리를 충분히 분석 할 수 있겠지만 잘 모른다면 Git을 얼마나 잘 사용하고 있는지에 대한 정보가 부족할 수 있다. 이를 해결하기 위해 Git inspector에 기능을 추가해 평가 요소를 제공하는 것을 고려할 수 있다.
2. 레포지터리를 pull 하는 것이 어느 정도의 시간이 걸릴지 체크해 봐야 한다. 현재는 따로 DB를 두지 않고 S3에 저장되어 있는 정보를 요청이 들어왔을 때 AWS Lambda를 통해서 pull을 해온 후에 유저가 지정해 준 조건에 맞추어 웹에 나타내주는 방식으로 설정했다. 하지만 만약 pull 역시 시간이 오래 걸린다면 따로 DB를 두어서 거기에 pull 정보를 담아놓고 유저의 요청이 들어왔을 때 단순히 DB에서 정보를 꺼내어 웹으로 나타내 주는 방식도 있을 것이다. 하지만, 이렇게 되면 유저의 조건을 받아들이기에 어려움이 있기 때문에 가능하면 현재 방식이 좋을 것이라고 생각한다.

# **프로젝트 팀 구성 및 역할 분담**

**프로젝트에 참여하는 멤버의 역할을 구체적으로 명시한다.**

| 이름 | 역할 |
| --- | --- |
| 전호현 |  |
| 이근하 |  |
| 유문상 |  |
| 최원대 |  |

# **프로젝트 비용**

|  |  |
| --- | --- |
| **항목** | **예상치 (MD)** |
| 아이디어 구상 | 3 |
| 관련 정보 수집 | 10 |
| 프로젝트 관련 문서작업 | 5 |
| Back-end 구축 | 30 |
| Frond-end 구축 | 30 |
| 최종 테스트 | 8 |
| 합 | 86 |

# **개발 일정 및 자원 관리**

## 개발 일정

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **항목** | **세부내용** | **2월** | **3월** | **4월** | **5월** | **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

|  |  |  |  |
| --- | --- | --- | --- |
| 마일스톤 | 개요 | 시작일 | 종료일 |
| 계획서 발표 | 프로젝트 아이디어 선정  프로젝트 아키텍처 구상  관련 정보 수집 및 연구  **산출물 :**   1. 프로젝트 수행 계획서 2. 프로젝트 소개 ppt | 2018-03-02 | 2018-03-09 |
| 중간 보고 | Back-end 구축 완료  Git inspector를 통한 간단한 결과 페이지  **산출물 :**   1. 프로젝트 중간 보고서 2. 중간 소스 코드 | 2018-03-10 | 2018-04-12 |
| 구현 완료 | Front-end 구축 완료  **산출물:**   1. 완전한 웹 어플리케이션 2. 최종 소스 코드 | 2018-04-13 | 2018-05-22 |
| 테스트 | 시스템 통합 테스트  **산출물:**   1. 버그 조사서 | 2018-05-23 | 2018-05-28 |
| 최종 보고서 | 최종 보고  **산출물:**   1. 최종 보고서 2. 웹 어플리케이션 | 2018-05-29 | 2018-05-29 |

## 인력자원 투입계획

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 이름 | 개발항목 | 시작일 | 종료일 | 총개발일(MD) |
| 전호현 | *Git inspector 기능 개선과 웹 제작* | 2018-03-02 | 2018-05-22 | 30 |
| 유문상 | *AWS lambda, Amazom S3 등을 이용한 Back-end 구성* | 2018-03-02 | 2018-05-22 | 30 |
| 이근하 | *AWS lambda, Amazom S3 등을 이용한 Back-end 구성* | 2018-03-02 | 2018-05-22 | 30 |
| 최원대 | *Git inspector 기능 개선과 웹 제작* | 2018-03-02 | 2018-05-22 | 30 |

## 비 인적자원 투입계획

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 항목 | Provider | 시작일 | 종료일 | Required Options |
| 개발용 노트북 | 개별 지참 | 2018-03-02 | 2018-05-29 |  |
| Amazon S3 | Amazon | 2018-03-02 | 2018-05-29 |  |
| AWS lambda | Amazon | 2018-03-02 | 2018-05-29 |  |

# **참고 문헌**

**참고한 서적, 기사, 기술 문서, 웹페이지를 나열한다.:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 번호 | 종류 | 제목 | 출처 | 발행년도 | 저자 | 기타 |
| 1 | 인터넷 | Github Webhooks | https://developer.github.com/webhooks/ |  |  |  |
| 2 | 인터넷 | AWS Lambda | https://aws.amazon.com/ko/lambda/features/ |  |  |  |
| 3 | 인터넷 | Amazon S3 | https://aws.amazon.com/ko/s3/ |  |  |  |