A1.2

프로젝트 계획서

|  |  |  |  |
| --- | --- | --- | --- |
| 일자 | Version | 설명 | 담당자 |
| 2011.04.10 | V1.0 | 최초 프로젝트 계획서 | 박민우 |
| 2011.06.15 | V2.0 | 프로젝트 계획서 수정 | 윤진경 |
|  |  |  |  |

A.1.2.1 프로젝트 타당성 조사

* 1. 프로젝트 타당성
  2. 차별성

A.1.2.2 **프로젝트 규모계획**

1. **규모산정 및 투입공수계산**
2. **프로젝트 일정계획표**

A.1.2.3 **프로젝트 품질관리**

A.1.2.4 **프로젝트 위험관리**

A.1.2.1 프로젝트 타당성 조사

1. 프로젝트 타당성

오늘날 Facebook과 Twitter와 같이 인터넷 상에서 사람들과의 교류와 동시에 자신들의 인맥이나 정보 공유 등의 목적과 같은 소셜네트워크서비스에 대한 사용이 증가하고 있다. 더욱이 소셜네트워크서비스 상에서 자신의 홈페이지를 관리할 수 있을 뿐만 아니라 Twitter의 트윗이나 Facebook의 담벼락과 같은 공간에 자신의 일상생활이나 때로는 좋거나 혹은 안 좋은 감정들을 글을 통해서 기록을 남기고 있다.

본 프로젝트에서는 글쓰기 기능에 초점을 두어 문장 속에서 자신의 감정을 표현할 수 있는 감정동사나 감정형용사들을 인식하여 긍정`부정과 같이 감정 인식 후에 현 상태를 화면 UI나 구조 또는 테마를 바꿔줌으로써 직접 글을 읽기 전에 시각효과를 통하여 감정 전달을 할 수 있게 하려는 것이 본 프로젝트의 목적이다.

1. 차별성

본 프로젝트에서는 기존 Social Network Service 상에서 자신의 상태와 감정표현을 할 수 있는 글쓰기 기능에 초점을 맞추어 기존 서비스와 차별성을 두고자 한다. 사람들은 SNS 상에서 자신의 감정을 글을 통하여 많이 표현하고자 한다.

예를 들어, 상황에 따라 사람들의 감정은 좋을 때도 있고 나쁠 때도 있다.

본 프로젝트의 주 핵심은 우리가 작성하는 글 속에서 “감정동사”나 "감정 형용사"를 인식해 현재 나의 감정상태에 따라 작성된 글의 UI나 구조 혹은 테마 등을 변환시켜 주는데 의의를 두고 있다. 따라서 글을 읽어보지 않아도 색깔로서 나의 감정상태를 파악할 수 있는 장점이 있으며, 또한 현재 추세인 Social Network Service에 감성공학을 접목시켜 자신의 감정을 실시간으로 시각적으로 표현할 수 있는 방법에 대하여 연구하는데 기존 SNS 방식들과는 차별화된 본 프로젝트만의 독창성을 가진다.

A.1.2.2 **프로젝트 규모계획**

1. **규모산정 및 투입공수계산**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **작 업 명** | **산출물** | **가중치** | **공수(MH)** |
| **1** | **사전조사** | 개발방법론, 프로젝트계획서 | 0.1 | 102 |
| **2** | **문제분석** | 문제기술서 | 0.1 | 102 |
| **3** | **요구분석** | Use Case Diagram  Use Case Spec | 0.3 | 338 |
| **4** | **설 계** | 프로그램 기능목록  클래스 목록  클래스 명세  Sequence Diagram  Class Diagram  화면  ERD | 0.5 | 482 |
| **5** | **구현** | 소스코드 | 1 | 1,024 |
| **6** | **테스트** | 테스트결과서 | 1 | 1,024 |

**규모산정 근거**

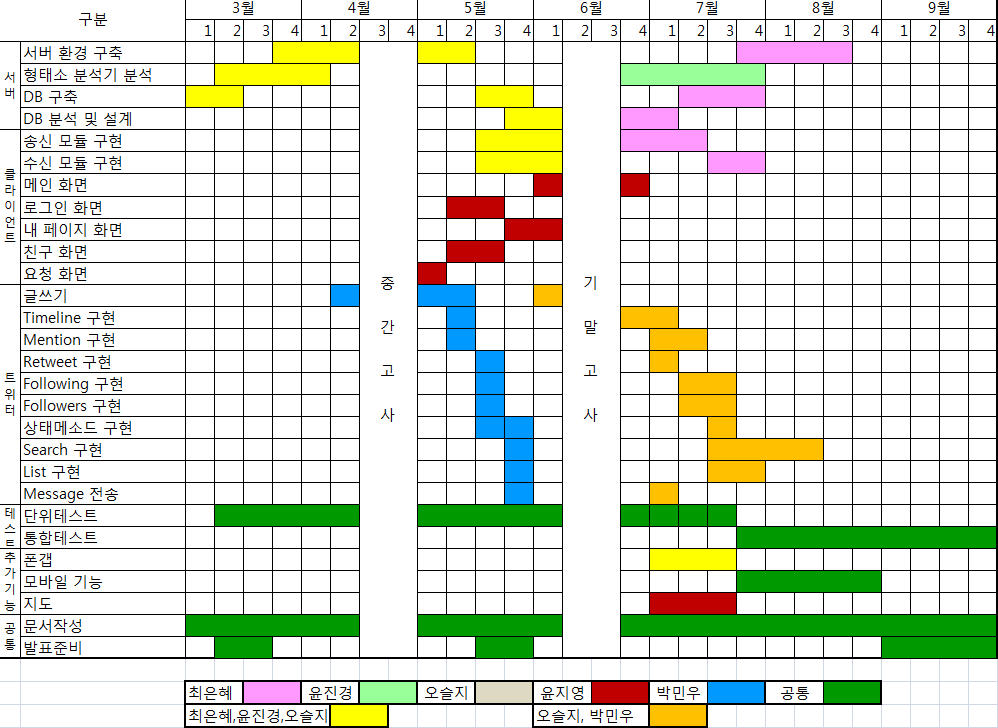
MH를 이용하여 추정, 구현계획을 참조하면 됨.

1024(MH) X 0.1 = 102(MH) (계획단계는 사전조사, 문제분석, 요구분석, 설계를 포함하고 전체 MH에서 각 단계의 가중치를 곱해서 MH를 산정한다.)

테스트는 모든 프로젝트의 전 과정에서 이루어져야 하기 때문에 구현단계의 공수와 같다고 가정

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 작 업 | 상 세 | 구현코드  규모산정(LOC) | 공수(MH) | 오픈소스 분석 공수(MH) | 총(MH) |
| 서버 | **서버환경 구현** | 600 | 30 | 0 | 30 |
| **DB 구축** | 500 | 25 | 0 | 25 |
| **DB 분석 및 설계** | 100 | 5 | 0 | 5 |
| 클라이언트 | **송신 모듈 구현** | 400 | 20 | 0 | 20 |
| **수신 모듈 구현** | 400 | 20 | 0 | 20 |
| **메인 화면** | 200 | 30 | 0 | 30 |
| **로그인 화면** | 100 | 15 | 0 | 15 |
| **내 페이지 화면** | 300 | 50 | 0 | 50 |
| **친구 화면** | 300 | 50 | 0 | 50 |
| **요청 화면** | 100 | 15 | 0 | 15 |
| 트위터  연동 | **글쓰기** | 600 | 55 | 10 | 55 |
| **timeLine구현** | 500 | 20 | 10 | 20 |
| **Mention 구현** | 200 | 20 | 10 | 20 |
| **Retweet 구현** | 300 | 30 | 5 | 30 |
| **Following 구현** | 200 | 20 | 5 | 20 |
| **Followers 구현** | 100 | 20 | 5 | 20 |
| **상태메소드 구현** | 400 | 33 | 5 | 33 |
| **Serch 구현** | 200 | 10 | 5 | 10 |
| **List 구현** | 200 | 10 | 5 | 10 |
| **Message 전송** | 600 | 20 | 5 | 20 |
| 오픈소스 | **형태소 분석기 분석 및 수정** | 5500 | 80 | 20 | 80 |
| Total | | 11800 | 578 | 85 | 578 |

1. **프로젝트 일정계획표**



A.1.2.3 **프로젝트 품질관리**

프로젝트가 원활한 품질 표준을 따르는지 결정하기 위한 프로젝트의 과정을 살펴본 뒤 불만족스러운 수행의 원인을 찾은 뒤 이를 제거하는 방법을 말한다.

프로젝트가 수행해야 할 요구를 만족시킨다는 것을 보증하는 프로세스들을 포함한다.

프로세스의 산출물을 면밀히 검토하여야 하며, 검토결과 발생하는 문제점을 수정하는 식으로 진행 해야 한다. 또한 문제점에 대한 팀원간의 해결 방안에 대한 의견을 적절히 조율하여 프로젝트의 근본적인 문제점을 해결하도록 하는 것이다.

|  |  |
| --- | --- |
| **Quality Planning** | 품질 표준을 정하고, 그것을 만족시킬 방법을 결정 |
| **Quality Assurance** | 프로젝트가 품질표준을 만족시킬 것이라는 확신을 제공하기 위해 정기적으로 전반적인 프로젝트 성과를 측정 |
| **Quality Control** | 프로젝트 결과가 품질표준을 만족하는지 모니터링하고, 만족한 성과의 원인을 제거하는 방법을 정의 |

이 프로세스들은 서로 상호작용하며 다른 관리영역에 있는 프로세스들과도 그러하다. 각 프로세스는 프로젝트 요구에 근거하여 한 명 이상의 투입이 필요하다. 각 프로세스는 모든 프로젝트 단계에서 적어도 한 번은 발생한다. 프로젝트 품질관리는 Project 관리와 Product 양쪽 모두를 충족해야 한다. 이 두 가지 측면에서 요구되는 품질을 만족시키지 못할 경우 프로젝트 이해관계자들에게 부정적 인식을 초래할 수 있다.

|  |  |
| --- | --- |
| **고객 만족** | 고객의 기대를 만족시키기 위해서 요구를 이해하고, 관리하고, 영향력을 발휘하며 이것은 요구사항에 맞게 그대로 만들어진 제품과 제품이나 서비스가 요구하는 것을 만족하는 것의 조화를 요구 |
| **검사를 통한**  **사전 예방** | 오류를 수정하기 위한 비용보다 이를 방지하기 위한 비용이 훨씬 적게 듦 |
| **관리 책임** | 프로젝트 성공은 모든 팀원의 참여가 필요하지만, 이것은 필요한 자원을 공급해야 하는 관리층에 책임을 남김 |

A.1.2.4 **프로젝트 위험관리**

**1. 위험관리 계획**

: 프로젝트를 실패하게 하는 위험요소들을 미리 식별하고, 그 해결방안을 모색하여 프로젝트를 진행하는 과정에서 계속적으로 위험을 관리한다. 위험 정도에 따라 대, 중, 소로 나눌 수 있다.

가) 위험 정도 ‘대’: 프로젝트 진행에 큰 영향을 주어 프로그램이 실패로 이어질 가능성이 높은 위험요소

나)위험 정도 ‘중’: 위험가능성은 있지만, 발생요인이나 시간 등을 예측할 수 있기 때문에 해결이 가능

다)위험 정도 ‘소’: 진행상 위험 가능성이 있지만 그 정도가 미약하고 각 담당자선에서 해결 가능한 위험요소

**2. 위험 식별 및 대처방안**

: 위험의 종류는 4가지로 나눌 수 있다. 기술적 위험, 스케줄 위험, 잘못된 기능/성능의 소프트웨어 개발 위험과 종합적 위험으로 나뉜다. 각각의 위험종류에 따른 대처방안을 적용한다.

가) 기술적 위험(위험 정도 ‘대’): 요구되는 기술수준의 미흡에 따른 업무수행 실패

→ 부족한 기술은 참고서적과 인터넷을 이용하고, 주말마다 팀원 간 스터디와 전문가의 도움을 받는 등의 방법으로 해결한다.

나) 스케줄 위험(위험 정도 ‘중’): 예상 기간 내에 일을 완료하지 못하게 되는 요인과 개인 사정으로 인해 특정 팀원이 역할을 수행하지 못하게 되는 요인

→ 각각의 임무에 대한 숙지와 팀원 간 의사소통을 원활히 하여, 만약 기간을 지키지 못할 경우 팀원간 토론을 통해 문제점을 모색하여 함께 해결한다. 특히, PM의 지속적인 관리가 필요하다.

특정 팀원에게 개인사정이 생겨서 업무를 완료할 수 없을 경우를 대비하여, ‘정’, ‘부’의 개념을 두어 커버한다.

|  |  |
| --- | --- |
| 정 | 부 |
| 최은혜[DB,연결] | **윤진경** |
| 윤진경[형태소 분석기] | **최은혜** |
| 박민우[트위터 연동, 클라이언트] | **오슬지** |
| 오슬지 [트위터 연동,모바일] | **박민우, 윤지영** |
| 윤지영[클라이언트, 모바일기능] | **박민우** |

다) 잘못된 기능/성능의 소프트웨어 개발 위험 (위험 정도 ‘소’ )

→ 관련된 기능을 가진 자료를 참고하여 문제점을 분석하여 수정범위를 찾아낸다. 수정이 이루어질 경우 결과는 산출물에 기록한다.

라) 종합 위험 (위험 정도 ‘대’)

: 위의 위험요소가 종합적으로 나타낼 때

→ 일단 문제가 무엇인지 파악하고, 관련된 참고자료를 찾는다. 기술적 위험과 스케줄 위험, 기능 성능의 위험요소들 중에 어떤 위험요소가 가장 큰 문제인지 파악한다. 그리고 그것에 맞추어 위의 대처방안을 적용한다. 이런 종합적인 위험은 팀 차원의 문제로서 접근하고, 각 분야별로 나누어진 위험요인들을 각 담당자 별로 해결방안을 모색한다.

하지만, 무엇보다도 이러한 위험이 발생하지 않도록 지속적인 회의를 한다. 이를 통해 문제점을 바로 찾아내서 위험에 대처하도록 하는 것이 중요하다.