

**Software Engineering Class01 term project**

**SimpleMerge Team7**

**Project Management Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Team Leader 김정민 |  | Team Member  김상헌, 양종만, 이동현, 임기찬, 황선준 |  | Writer  김정민 |

**COTENTS**

1. 프로젝트 [개요](#_Toc349734314) 2
2. [프로젝트 진행 3](#_Toc349734319)
3. 프로젝트 후기 [6](#_Toc349734331)

# 프로젝트 개요

* Repository Address

<https://github.com/minn951120/SimpleMerge>

* 프로젝트 관리자

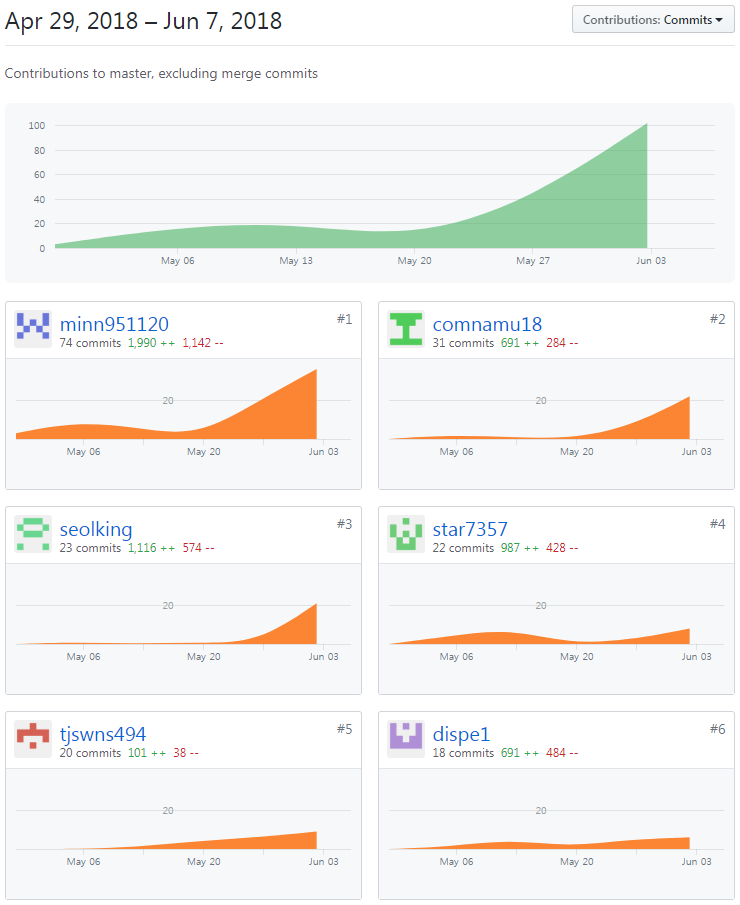
김정민

| **구성** | **이름** | **역할** |
| --- | --- | --- |
| 팀장 | 김정민 | Controller, Code Review, Project Manager |
| 팀원 | 김상헌 | Model, SRS, demo presentation, quality assurance |
| 팀원 | 양종만 | Model, LCS, test/design/analysis/implementation docs |
| 팀원 | 이동현 | Model, SRS |
| 팀원 | 임기찬 | View, Controller, test/design/analysis/implementation docs |
| 팀원 | 황선준 | View, presentation, design/analysis/implementation docs |

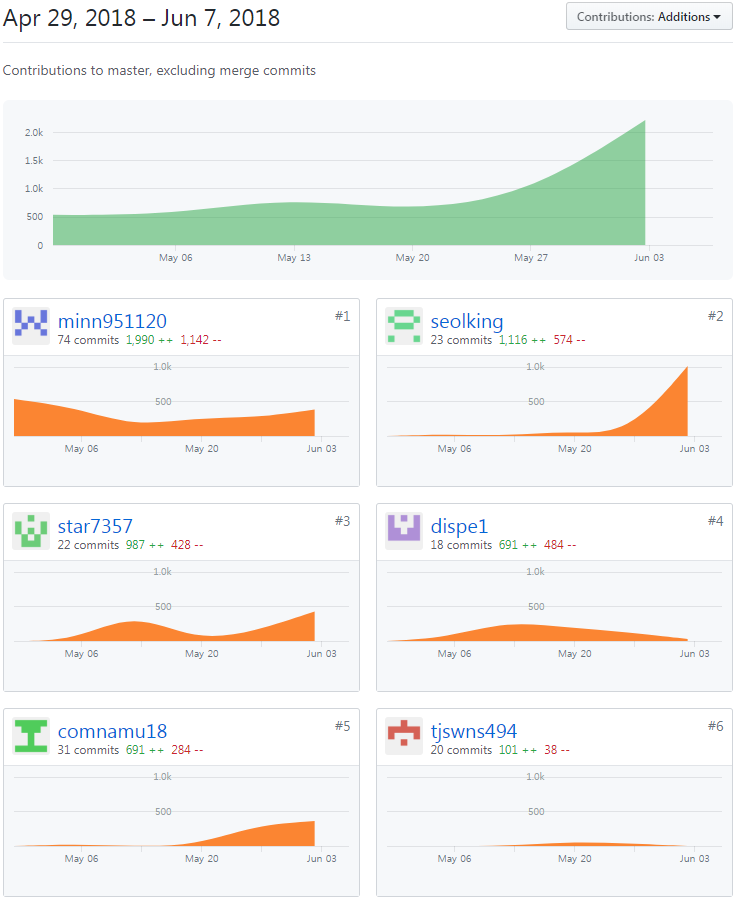
* 프로젝트 멤버와 역할
* License : MIT

# 프로젝트 진행

* **개요**
  + 350 commits
  + 6 contributor (모든 팀원 기여)
  + 각자 repository를 fork하여 작업 후 pull request로 original repository에 commit
* **Contributing Insight in GitHub**
  + Commits



* + additions



* **주별 진행**

약 1주일의 주기로 모든 팀원들이 모여 디자인에 대한 회의를 하고, 프로그램의 각 부분(Model, View, Controller) 별로 구현을 진행하였습니다.

* + 5월 3일
* GitHub repository 생성, 기본 설정
* 역할 분배
* Major Use Case Design
* 일정 계획
* 개발환경 구축
* SRS 작성
  + 5월 14일
* Load 기능, View prototype구현
* 프로그램의 흐름(행동 제약) 구체화
  + 5월 17일
* Model design, prototype 구현
* View css 적용, ui design 확정
* Save, edit, compare 기능 구현
* Contoller, View 역할별로 분리
  + 5월 28일
* Binding 구체화
* 기능상의 버그 수정
* 일정 연기
* Model code refactoring
* design문서, SRS 작성
  + 6월 4일
* copy 기능 구현
* 기능상의 버그 수정
* 각종 문서 작성 및 최종 제출

# 프로젝트 후기

* Git, GitHub를 통한 버전관리, 프로젝트 관리

프로젝트 초기에는 팀의 대부분의 멤버가 git, GitHub의 사용에 익숙하지 않았습니다. Repository, staging area와 같은 개념에 익숙해지고 제대로 사용하기까지 예상보다 긴 시간이 소요됐고 그로 인해 초기 개발 진척이 더딘 면이 있었습니다. 익숙해진 후에는 각자 맡은 부분의 구현을 병렬적으로 수행할 수 있게 되어 속도를 낼 수 있었습니다.

다른 팀원들의 개발 진행 사항을 실시간으로 확인하여 피드백하고 의사소통을 할 수 있는 부분이 좋았습니다.

Branch, revert와 같은 기능을 잘 사용하지 못해 여러 기능을 동시에 작업할 경우, 코드를 복원해야 할 경우에 불편함이 있었습니다. Git의 유용한 여러 기능들에 대한 개념적인 공부가 필요할 것 같습니다.

* 설계, 문서화를 병행한 협업 개발

MVC 패턴을 사용하여 개발함으로써 모듈을 분리하고 팀원들이 각자 맡은 부분을 구현함으로써 병렬적으로 코드 작성을 할 수 있었습니다. 문서를 통해 프로그램의 제약사항, 디자인을 기록해 둠으로써 구현 중에 의사소통에 드는 비용을 감소시킬 수 있었습니다.

설계에 대한 지식과 경험이 모자란 상태에서 시작한 프로젝트였기에 부족한 부분도 많았습니다. Interface를 정의하지 않음으로 각 묘듈(특히, Controller와 Model) 사이의 의사소통을 위해 소요된 시간이 너무 많았습니다.