**<GitClass> Release Summary (5 pages MAX)**

**Team members**

|  |  |  |
| --- | --- | --- |
| Name and Student ID | GitHub ID | Number of story points that member was an **author** on |
| **Ivan Bogdev** | 16981596 | 14 story points |
| Yousef Abo El Foul | 25316395 |  |
| Jeffrey Buonamici | 11137499 |  |
| Mathiaco-Lee Bessane | 25252955 |  |
| Emanuel Mateus | 15270310 |  |
| Aghiles Sait | 9168157 | 5 points |
| Nathan Shummoogum | 19209440 |  |

**Project summary (max one paragraph)**

Our website was created to facilitate the communication between students and TAs in the Software Engineering and Computer Science department. It will provide a platform where students will be able to work and collaborate on their coding projects. It will also provide easily accessible GitHub statistics to make grader faster for the teachers. Students will be able to login directly through their GitHub account. Teachers can create classes. Students will be able to form groups within those classes, where they will be able to interact with each other, upload documents and have it all linked to their GitHub repositories. It will also offer a chat, allowing one to contact classmates and the TA for any questions. Our website will allow group projects in software engineering to be completed using only this one platform.

**Velocity and a list of user stories and non-story tasks for each iteration**

(make sure the iteration is clickable link to the milestone on github)

Total: 19 stories, 56 points over 10 weeks

Iteration 1 (X stories,  X points) <https://github.com/ivanb7/soen341group3/milestone/5>

US #14: Login GitHub [3 points] [Status: Done]

US #17: Access GitHub activities [ 3 points] [Status: Pushed]

Iteration 2 (X stories, X points) <https://github.com/ivanb7/soen341group3/milestone/6>

US #18: Chat system [5 points] [Status: Split]

US #16: Display profile page [ 3 points] [Status: Pushed]

US #22: Create a class [3 points] [Status: Pushed]

US #15: Evaluate Participation [8 points] [Status: Splitted]

Iteration 3(X stories, X points) <https://github.com/ivanb7/soen341group3/milestone/7>

US #18: Chat system [5 points] [Status: Splitted]

US #19: Message Group [5 points] [Status: Splitted]

US #26: Post Class [3 points] [Status: Done]

US #27: Upload document to group [ 3 points] [Status: Pushed]

US #22: Create a class [3 points] [Status: Done]

US #34: Control privileges [1 point] [Status: Done]

US #15: Evaluate Participation [8 points] [Status: Splitted]

Iteration 4(X stories, X points) <https://github.com/ivanb7/soen341group3/milestone/8>

US #35: Join class [2 points] [Status: Done]

US #18: Chat system [5 points] [Status: Done]

US #15: Evaluate Participation [8 points] [Status: Done]

US #19: Message Group [5 points] [Status: Done]

US #27: Upload document to group [ points] [Status: Done]

US #33: Save contacts [2 points] [Status: Removed]

US #38: #Post users made [1 points] [Status: Removed]

Iteration 5, Release (X stories, X points) <https://github.com/ivanb7/soen341group3/milestone/9>

US #36: Join groups [ 2 points] [Status: Done]

US #37: Post Group [5 points] [Status: Done]

US #28: Comment on users’ post [1 points] [Status: Done]

US #24: Create group [1 point] [Status: Done]

US #21: Upload document to chat [5 points] [Status: Removed]

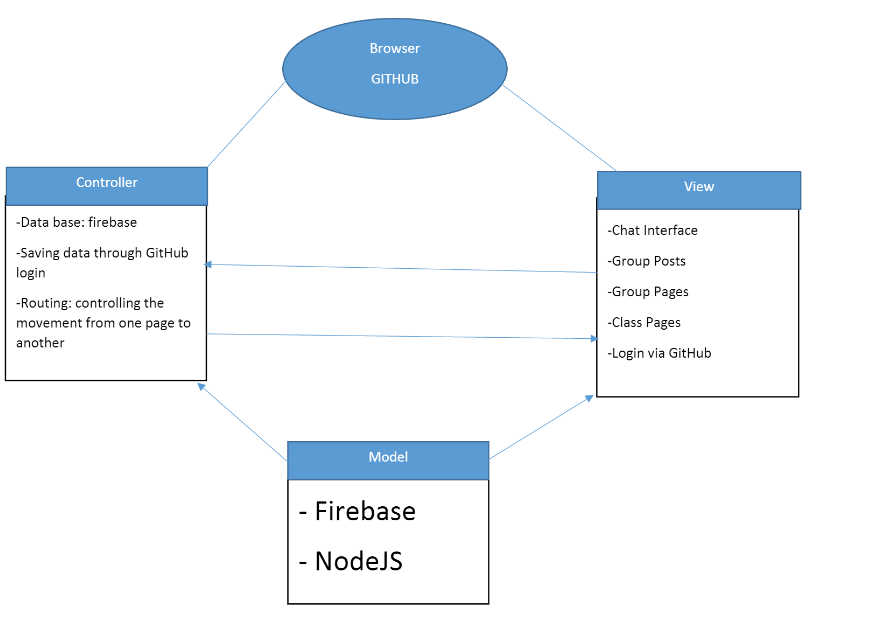
US #25: Subgroups [0 points] [Status: Removed]

**Overall Arch and Design**

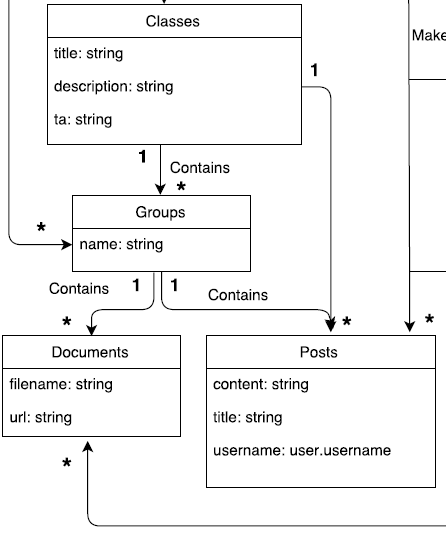
Show us the overall architecture in your system with architecture diagram.

Show applicable parts (at least one) from the 4+1 logical/physical/etc. model, with appropriate UML techniques we covered. You can also include these diagrams in your stories on GitHub (by providing urls).

Architecture: Model view controller

****

Applicable part:

****

**Infrastructure**

For each library, framework, database, tool, etc

Node.js

Firebase

Github

Heroku

Jquery

Travis CI

**Name and link**

Max 1 paragraph description of why you are using this framework.

Max 1 paragraph description of other alternatives and why they don’t work.

**Name Conventions**

*List your naming conventions or just provide a link to the standard ones used online.*

*For example:* [*Java naming conventions*](http://www.oracle.com/technetwork/java/codeconventions-135099.html)

**Code**

*Key files: top* ***5*** *most important files (full path). We will also be randomly checking the code quality of files. Please let us know if there are parts of the system that are stubs or are a prototype so we grade these accordingly.*

|  |  |
| --- | --- |
| File path with clickable GitHub link | Purpose (1 line description) |
| 1. |  |
| 2. |  |
| 3. |  |
| 4. |  |
| 5. |  |
|  |  |

**Testing and Continuous Integration**

*Each story needs a tests before it is complete. If some class/methods are missing unit tests, please describe why and how you are checking their quality. Please describe any unusually aspects of your testing approach.*

List the **5** most important unit test with links below.

|  |  |
| --- | --- |
| Test File path with clickable GitHub link | What is it testing (1 line description) |
| 1. |  |
| 2. |  |
| 3. |  |
| 4. |  |
| 5. |  |
|  |  |

List the **5** most important acceptance tests with links below.

|  |  |
| --- | --- |
| Test File path with clickable GitHub link | Which user story is it testing (1 line description) |
| US#18 | The chat system |
| US#27 | Uploading files |
| US#22 | Creating a class |
| US#17 | Showing graph of each member of their contribution to the project |
| US#37 | Adding classmates into the group and discuss via posting on the group forum |

For all the acceptance tests, the link is <https://github.com/ivanb7/soen341group3/blob/master/Documentation/Acceptance%20Tests>

*Describe your continuous integration environment. Include a link to your CI.*

*Describe the choice of the static analysis tool and how do you run it. The static analysis tool should analyze the language that are used in the majority of the your source code.*

The static analysis tool chosen to analyze the majority of the source code in this project is the online tool JSLint. This tool seemed to be a good choice as the principal language in the source code is node JS. With JS Lint, anomalies or defects in the source code were identified and corrected. To run the analysis, each file from the source code is pasted in [www.jslint.com](http://www.jslint.com) and the report on the code will be available. Appendix A is a screenshot of the analysis of the chat.js file.

*Attache a report as appendix from static analysis tool by running the static analysis tool on your source code.*

