Project Release Summary

GitHub repository monitoring tool

(https://github.com/abhandal/SOEN341-G4)

Team:

Name	Student id	GitHub id	Number of story points that member was an author on
Charles-Philippe Labbé	40002442	**CharlesPhilippeLabbe**	
Aman Bhandal	27390858	**abhandal**	
Dmitry Kryukov	40029645	**b5n**	
Ksenia Popova	40029623	**Lyncis**	
Nikita Baranov	40012854	**NikitaBaranov**	
Batoul Yehia	40010912	**batoulyehia**	
Raymart De Guzman	40010443	**tramyardg**	
Andy Pham	40006071	**andypham29**	

Project summary:

The GitHub repository monitoring tool helps the teaching assistants to monitor and grade each group and student, by analyzing a repository information in an easy-to-use and easy-to-understand way.

Main Features:

- Secure login with GitHub account
- Get an information from available repositories
- Show the information in a convenient way (text, charts ...):
    - Contributors (students)
        - a number of events: overall and per student
        - a percentage of events per student
    - Events:
        - Commits
        - Pull requests
        - Issues
        - Comments
- Write a feedback from TA that will be stored in teams' repositories

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

Total: X stories, X points over X weeks
Iteration 1 (X stories, X points)

US #X: US name [X points] [Status: Done, Removed, Pushed, or Splitted]

Iteration 2 (X stories, X points)

- Log In (3)
- Events Separation (3)
- Adding repositories (2)
- Generation Information (3)
- Overall Report
- Function of Activity Percentage (2)
- Tabs
- Data Selection (1)

Iteration 3(X stories, X points)

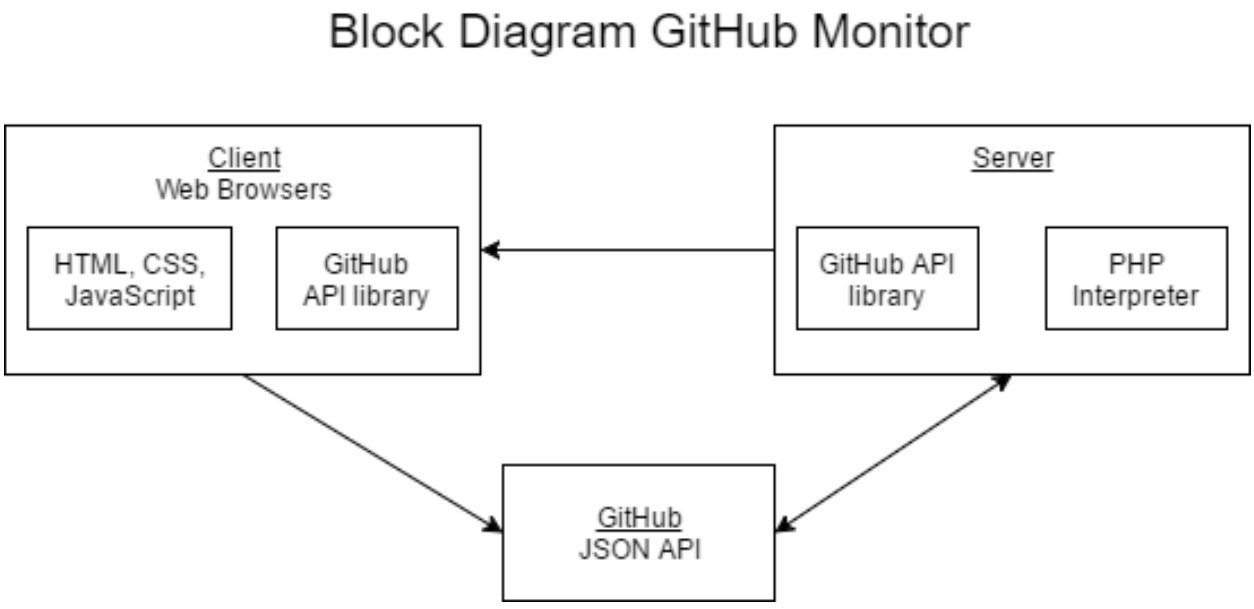
- Burndown Chart (?)
- Comments (3)
- Selecting Repository (5)
- Sorting the information of the user interface
- Weekly Report (3)
- Implement a graphing tool (Coffee)
- Report Type

Iteration 4(X stories, X points)

Iteration 5, Release(X stories, X points)

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).



Infrastructure

For each library, framework, database, tool, etc

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.

Server side:

- PHP Interpreter

- PHP GitHub API 2.0 (https://github.com/KnpLabs/php-github-api)
  A simple Object Oriented wrapper for GitHub API, written with PHP5
- Amazon Web Services (AWS) (https://aws.amazon.com)
  Free of charge reliable web hosting.

Client side:

- HTML, CSS (https://www.w3.org/html/)
- JavaScript (http://www.ecmascript.org)
- Chart.js (http://www.chartjs.org)
- JQuerry.js (https://jquery.com)
- Material Design (https://material.io)
- Bootstrap (http://getbootstrap.com)
- Github.js (https://github.com/github-tools/github)

Automation testing: Travis CI.

Name Conventions

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

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)
Integration/js/repo.js https://github.com/abhandal/SOEN341-G4/blob/master/Integration/js/repo.js	Core Data extracting object.
frontend/repo_selection.html https://github.com/abhandal/SOEN341-G4/blob/master/frontend/repo_selection.html	Main application page
PHP login page	Login and security management

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)

https://github.com/abhandal/SOEN341-G4/blob/master/test/unit/gh_api_test.js	
https://github.com/abhandal/SOEN341-G4/blob/master/test/unit/repoStub.js	
https://github.com/abhandal/SOEN341-G4/blob/master/test/unit/repoStub_test.js	
https://github.com/abhandal/SOEN341-G4/blob/master/test/unit/repoTest.js	
https://github.com/abhandal/SOEN341-G4/blob/master/test/unit/test.html	

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

Test File path with clickable GitHub link	Which user story is it testing (1 line description)
https://github.com/abhandal/SOEN341-G4/blob/master/test/integration/login_test.js	
https://github.com/abhandal/SOEN341-G4/blob/master/test/integration/get_data_test.js	
https://github.com/abhandal/SOEN341-G4/blob/master/test/integration/get_repo_commits_test.js	
https://github.com/abhandal/SOEN341-G4/blob/master/test/integration/get_repo_collaborators_test.js	

List the **5** most important Acceptance tests

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.

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

Challenges

Aman Bhandal: The main challenge areas involved setting up the testing environment and (insert bs here)

Raymart De Guzman: I faced are two different challenges during the course of the project.

First, learning Material Design Lite (MDL) framework for the front-end development. This is framework is all new to me. However, I did quite a bit of work in Bootstrap, a framework similar to MDL.
I've been using Bootstrap in my own projects. So, this was not a surprise for me.
It took me a couple of hours to learn the basics and finally applied them to the project.
Second, understanding and learning about web application programming interface (API) especially GitHub API.
GitHub API is the main web API used in the project, thus, I had to read the API documentation and make sense
how are we going to employ their usage in our project. Fortunately, playing around with GitHub's OAuth authorization
API helps me understand API application better, as well as its relevance
in other projects.

Ksenia Popova: I met a few challenges during this project.
1. Work with Promises when I needed to use the data from them and it was not visible outside Promises.
2. Chart type switch, because I needed to destroy the existing chart before that, but charts plugin.
3. Sorting of the data, because it is represented in object with different properties and it is not possible to make data sorted with one of the object properties.

Dmitry Kryukov: In this project the main challenges involved experience with asynchronous javascript
and also work with selenium webdriver library for creating integration tests.

Feedback

Ksenia Popova: This project was a very valuable experience for me for many reasons. First of all, I improved my skills in Javascript. My task was to display the data from the GitHub with charts and tables, so I needed to figure out how to do it and create some algorithms. Secondly, the project idea was to apply Scrum model to our development process, and this should be a very useful experience in the future. And finally, we have a great team. We had a lot of meetings and communicated with Slack, so we always stayed in touch. Everyone could get help with any part of the project and everyone did what he was supposed to do. We completed everything on time without any conflicts or delays, what is why I am very satisfied with this project.

Appendix 1 – Final Acceptance Test table.

Project:	GitHub repository monitoring tool	Browser:	Chrome
Written By:	Nikita Baranov	Description:	Full User story test
Tested By:	Aman Bhandal	Tested On:	12-Apr-17

#	Date	User Story	Expected Results	Actual Results	Pass
login to the System					
1	12-Apr	Log In	Should get to home screen using Github credentials	User logged in using GitHub credentials	Yes
2	12-Apr	Selecting Repository	Look throught available repositaries, select some of them to be available on system.	User got a list of available repositaries. Selected 2 and got them available on left panel of the system.	Yes
3	12-Apr	Adding repositories	Add repository to be availiable in the system.	User selected repository to be added and got them available on left panel of the system.	Yes
Overall functionality					
4	12-Apr	Tabs	The information should be presented in different tabs	The system has tabs for different reports	Yes
5	12-Apr	Generation Information	The system should provide the general information about the repository.	General info seen on the General InfoTab	Yes
6	12-Apr	Overall Report	The system should provide the overall report of the repository	Owerall report is on the Overall Report Tab	Yes
7	12-Apr	Weekly Report	The system should provide the weekly splits of overall report of the repository	Owerall report is on the Weekly Report Tab	Yes
8	12-Apr	Burndown Chart	The system should provide the burndown chart for the repository	Burndown Chart is on the Burndown chart Tab	Yes
9	12-Apr	Comments	Write a comment and save it to the abserved repository.	Wrote a comment and save it to the abserved repository. The page did not refreshed.	Yes
The functionality of the Report Tabs					
10	12-Apr	Sorting the information of the user interface	Table with information could be sorted in descending order	The information cold be sorted by clicking on the Sort button.	Yes
11	12-Apr	Function of Activity Percentage	The information should be seen in precentage	The information in tables are shown with precentages.	Yes
12	12-Apr	Events Separation	The information should be separated to commits, issues, and comments.	Different events are shown separately.	Yes
13	12-Apr	Implement a graphing tool	The information should be presented in graphical view	There is Donut, Pie, Line and Bar charts.	Yes
14	12-Apr	Report Type	The required information should be displayed. In table forms or charts, or both combined.	The table and Charts are on the pages.	Yes
15	12-Apr	Data Selection	The system should display only selected information.	The information could be chosen using checkboxes on the top of the page.	Yes

Appendix 2 – User Story Screen Shots

1. Log In
   As a user, I want to log in using my GitHub account.
2. Adding repositories
   As a user, I want be able to add repositories on my page, so the data from it can be displayed.
3. Selecting Repository
   As a user, I want to select a repository from a GitHub account.
4. Generation Information
   As a user, I would like to have a general information page about a repository.
5. Comments
   As a user, I want to add comments to each repository.
6. Sorting the information of the user interface
   As a user, I would like to sort the charts and the tables from the least commits to the most.
7. Overall Report
   As a user, I would like to have an overall report of all the collaborators' activities.
8. Weekly Report
   As a user, I would like to see a weekly report of all the collaborators' activities.
9. Function of Activity Percentage
   As a user, I would like to see the percentage of each collaborator's activity.
10. Events Separation
    As a user, I would like to see a separation of events to commits, issues, and comments.
11. Burndown Chart
    As a user, I would like to have access to the burndown chart of the team.
12. Tabs
    As a user, I want to see different tabs in order to have access to general information, reports, burndown charts, and comments.
13. Implement a graphing tool
    As a user, I would like to see the breakdown of events into tables and charts.
14. Report Type
    As a user, I want to choose how the required information will be displayed. In table forms or charts, or both combined.
15. Data Selection
    As a user, I want to be able to select the data shown to me.

Old Report

User Stories

Extra functions

1. Periodical Email Summaries
   As a user, I want the system to send me an email with an overall summary report on activities in repositories so that I can spend less time to get an overall information on a class process.
2. Time Range
   As a user, I want to select the time range for the activities for the team and each person: I can select a starting date and an ending date.
3. Separation by sprints
   As a user, I want to specify sprints as a time range to be able group information in charts by sprints.
4. Grading space
   As a user, I would like to publish grades for each sprint so that the team could see it.
5. Repository Description
   As a user, I want to be able create my own description to each repository in my list, that I could recognize repository in my own association.
6. Managing repositories
   As a user, I should be able to manage repositories on my page, that I could edit list of repositories.

What should be in the doc (delete before submission)
TA meeting:
Final report: description, US, Architectural blocks and some feedback.

Course outline:
complete report integrating all sprints of the project,
demonstration of software, and quality of application