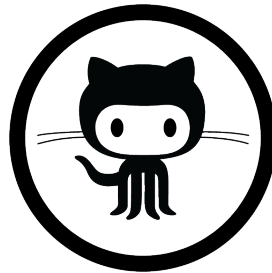




UNSW
A U S T R A L I A

SOFTWARE ENGINEERING WORKSHOP 1 (SENG1031)

GitHub and APIs and stuffff



Authors:

Daniel BU

z

Paul JUKIC

z

Robert MA

z

Joshua MORTON

z

Sanjay NARAYANA

z3424639

September 16, 2014

Contents

I	Project Scoping	3
1.1	GitHub - An Overview	3
2.1	Vision Statement	3
2.2	General Goals	3
2.3	Group/Team Goals	4
2.4	Individual Team Member Goals	4
3.1	Old Problem Statements	5
3.2	New Problem Statements	5
3.3	Explanation of Problem Statement Changes	5
3.4	Inference for Problem Statements	6
II	User Stories	7
1.1	User Stories	7

Phase I

Project Scoping

Deliverable 1

1.1 GitHub - An Overview

GitHub is a collaborative site for sharing and working on software projects based around the *Git Version Control System*. It is beneficial when used individually or as a group as it allows for members to use a web-based graphical interface to interact with the command line tool git.

GitHub's advantages are its ability to provide easy interaction between members working on the same project, and its ease of use compared to working from the command line. It allows for easy management of wikis, bug tracking and feature requests.

The GitHub system also contains features for sharing small code snippets (Gists) which act like normal repositories, so that they can be changed by the author (managed by Git version control on the website) and forked (making a separate copy of the project). Contributors are able to make changes to their projects locally and then push their changes to GitHub, but also can make changes directly on the online GitHub service as well.

Deliverable 2

2.1 Vision Statement

To create a website which interfaces with GitHub's API allowing for a more advanced searching of repositories and improving a user's experience of viewing a project; both in an holistic perspective and a detailed view. We also aim to link relevant sections of code with documentation from the wiki and allow for an easier workflow for the user.

2.2 General Goals

- Expand the *Explore GitHub* page and allow users to search for specific projects to increase contribution with a larger variety of open source projects.
- Provide and display relevant information and statistics in an intuitive manner to aid users with choosing new open source projects.
- Construct easy access between code and relevant documentation.
- Create a simple and convenient way to navigate through documents, files and folders; allowing for users to view and contribute to specific sections easier.

- Mediate a wider variety of resources (including statistics, code and projects) for utilisation.
- Create a holistic view of projects to understand how systems interact with one another, allowing for deeper learning in the workings of current and new projects.

2.3 Group/Team Goals

Statistics Team: (Josh & Daniel)

- Gather resources and statistics from current projects hosted on GitHub
- Working with the *Visual Team* to provide statistics which will shape the UI and workflow.

Visual Team: (Sanjay, Robert & Paul)

- Develop a seamless User Interface (UI) that flows from the existing, to allow easy access to important data.
- Creation of shortcuts to speed up common tasks (utilising data supplied from the *Statistics Team*).

2.4 Individual Team Member Goals

1. Josh - Browse multiple repositories on **GitHub** in order to devise a set of statistics which consider a project to be more readily contributable.
2. Daniel - Develop a system to aid users in selecting a project by displaying relevant statistics such as average input per contributor.
3. Paul - Develop a visual based system to manage and see the interactions between project files and other projects hosted on **GitHub**.
4. Sanjay - Implementing methods for connecting functions from user code to the wiki. Also to oversee development of our project documentation using \LaTeX .
5. Robert - To provide a more convenient way of viewing code by implementing the collapsing of functions/code blocks.

Deliverable 3

3.1 Old Problem Statements

At the current state, GitHub:

1. Does not have more advanced searching of repositories to find more readily contributable open source projects.
2. Does not display enough statistics (e.g. Average input per contributor) to aid with choosing a new project.
3. **Does not have a visual way to view a whole project (e.g. how all the classes/files interact, UML diagrams generated from code).**
4. Does not link documentation from a repository's wiki with the relevant functions/lines in the project's code.
5. **Does not allow functions/blocks of code to be collapsed for easy targeted viewing online.**

3.2 New Problem Statements

At the current state, GitHub:

1. Does not have more advanced searching of repositories to find more readily contributable open source projects.
2. Does not display enough statistics (e.g. Average input per contributor) to aid with choosing a new project.
3. Does not have a visual way to view a whole project (e.g. how all the classes/files interact, collapsing of functions/blocks of code).
4. Does not link documentation from a repository's wiki with the relevant functions/lines in the project's code.
5. Does not have the ability to search for users (contributors) based on how active they are and what kind of projects they contribute to.

3.3 Explanation of Problem Statement Changes

Problem statement 5 was found to be very specific and didn't allow enough room for creating features that could stem off of it. Also we thought of a new idea that GitHub did not have, the ability for authors' of repositories to search for other coders to contribute to their project. Subsequently, the old problem statement 5 was able to be easily merged with no. 3, giving it more breadth to allow for more features to be described.

3.4 Inference for Problem Statements

Inference for our problem statements stemmed from previous work and experience with GitHub. With a moderate level of knowledge of GitHub, we were able to identify a lack of features prevalent within usage of the system. Those of us who had lacked experience with usage of GitHub were quick to investigate and understand some important features to be missing.

Our first problem statement was brought up as a way to encourage younger or novice programmers to take part in contribution to open-source projects. Finding and aiding in projects that are large, cumbersome and complex are oft-not viable for programmers starting out in the field. Hence we proposed developing a more refined search feature that allows to find more readily contributable open source projects. Essentially, we would be enhancing resources available for the user.

The lack of a diverse set of statistics (such as average contribution per user) was a driving factor for our second problem statement, complementing our first problem statement. The availability of a greater set of statistical resources would be beneficial to aid individuals in finding suitable projects to work on. This was our inference for our second problem statement.

We also observed, explored and identified the absence of a visual aspect within the GitHub system. We realised the lack of a holistic view of code on GitHub. Previous work with software development prompted us to understand the need for a more visual aspect to viewing the relationships, connections and associations with other files in the repository.

The absence of a link between *code*, (be it the functions within the code or only certain lines of code) and its respective *documentation* was also another feature which we believe would be helpful. Prior work with coding and looking up relevant documentation for a function prompted us to list this fourth problem statement.

Our final statement was also another visual aspect. Once again, due to previous coding experience we realised the ability to collapse sections of code would provide a more pleasing, aesthetic look. As opposed to a large, clumsy, confusing code layout, we believe this feature would allow the end-user to focus their attention on a particular section.

All in all, we derived the essence of our problem statements based upon problems that we had previously encountered. We also formulated the problem statements for features and resources that we believed would enhance the user experience of the GitHub system.

Phase II

User Stories

Deliverable 1

1.1 User Stories

Requirement: Does not have more advanced searching of repositories to find more readily contributable open source projects.

Feature: An advanced search panel for finding repositories that I can commit to.

Connextra Feature:

AS A user (GitHub contributor)
SO THAT I can help other developers
I WANT TO Find repositories that I can contribute to

Scenario:

GIVEN: that I am on the main page
WHEN: I click on “Find Repos”
THEN: I can see a list of repositories with some basic stats next to each one

Feature: Filter repositories based on a number of parameters.

Connextra Feature:

AS A user (GitHub contributor)
SO THAT I can find more relevant repositories
I WANT TO filter a list of repositories

Scenario:

GIVEN: that I am on the main page
WHEN: I select some parameters and click on “Find Repos”
THEN: I can see a sorted list of repositories based on my parameters

Requirement: Does not display enough statistics to aid with choosing a new project.

Feature: Repositories lists the average amount of commits per contributor.

Connextra Feature:

AS A User (GitHub contributor)
SO THAT I can view other contributors motivation
I WANT TO Find repositories that I can contribute to

Scenario:

GIVEN: that I am viewing a repository's statistics
WHEN: I click on "Contributors"
THEN: I can view the average commits per contributor

Feature: Repositories graphs average stay time of contributors.

Connextra Feature:

AS A user (GitHub contributor)
SO THAT I know how much I need to contribute
I WANT TO View a Comprehensive list on average activity time

Scenario:

GIVEN: that I am viewing a repository's statistics
WHEN: I click on "Loyalty"
THEN: I can see whether other contributors stay long

Requirement: Does not link documentation from a repository's wiki with the relevant functions/lines in the project's code.

Feature: Code in repository opens a hyperlink to the complete wiki documentation.

Connextra Feature:

AS A Code viewer
SO THAT I I can read the explanation for written code
I WANT TO understand the inner-workings of raw code.

Scenario:

GIVEN: that I am viewing raw code in a repository
WHEN: I click on a function
THEN: I should be linked to specific section of wiki explaining the code.

Connextra Feature:

AS A GitHub contributor
SO THAT I I possess the tools to form the link between
raw code and documentation
I WANT TO set up a hyperlink for code viewers to understand my
written code.

Scenario:

GIVEN: that I am developing a wiki documentation
WHEN: I click on "Connect Code to Wiki"
AND: I specify which line I want to connect to
THEN: I should be able to set up a hyperlink to the wiki documentation.

Feature: A small tool tip summary of that data type is shown (showing its methods and attributes.)

Connextra Feature:

AS A Code viewer
SO THAT I I can see the data type's methods & attributes
I WANT TO understand the implementation of data types in written code.

Scenario:

GIVEN: I am observing the data type in the code
WHEN: I hover my mouse over a data structure
THEN: a tool-tip window show display, showing the
attributes and methods of the data type.

Connextra Feature:

AS A GitHub contributor
SO THAT I I possess the tools to develop a tool-tip activated
by mouse hover
I WANT TO construct a way for users to posses a greater understanding
of data structures in code.

Scenario:

GIVEN: I am developing a wiki documentation
WHEN: I click on “Make Tool-Tip”
AND: I specify which data structure I want to connect to
THEN: I should be able to develop a tool-tip explaining methods
& attributes of the data structure.

Requirement: Does not have a visual way to view a whole project.

Feature: Diagrams based on how files interact with one another

Connextra Feature:

AS A user (GitHub contributor)
SO THAT I can gain a deeper understanding of how the project
utilises other projects.
I WANT TO Find more projects linked to the current project.

Scenario:

GIVEN: that I am viewing a project or file
WHEN: I click on Model
THEN: I can get a diagram of which files utilise each other.
AND: I can click on the files to open them in a new window.

Feature: Holistic view of functions of code.

Connextra Feature:

AS A User (GitHub contributor)
SO THAT I can easily understand how a file works.
I WANT TO get a simplified version of the current code.

Scenario:

GIVEN: that I am viewing a file
WHEN: I click on “Collapse” next to a line of code
THEN: the code inside the function is hidden.
AND: is replaced with arrows showing where the function links to
WHEN: I click on “Collapse All”
THEN: the code inside all functions are hidden
AND: are replaced with a list using arrows to show where functions link to

Requirement: Does not have the ability to search for users (contributors) based on how active they are and what kind of projects they contribute to.

Feature: Searching for users based on how active they are.

Connextra Feature:

AS A User (GitHub author)
SO THAT I can collaborate with more active users
I WANT TO be able to search for users based on their average contributions

Scenario:

GIVEN: that I am on the main page
WHEN: I click on “Find Users”
AND: I select the parameter ”Sort by User Activity” and click “Search”
THEN: I can see a filtered list of users based on their average daily/weekly contributions

Feature: Searching for users based on what kind of projects they contribute to.

Connextra Feature:

AS A User (GitHub author)
SO THAT I can collaborate with users who have similar interests
I WANT TO be able to search for users based on the type of projects they’ve contributed to.

Scenario:

GIVEN: that I am on the main page
WHEN: I click on “Find Users”
THEN: I can open up the “Advanced Search”
WHEN: I select one of the project types under the “Project Categories” subheading
AND: I click “Search”
THEN: I can see a list of users who have contributed to any projects of the relevant type