

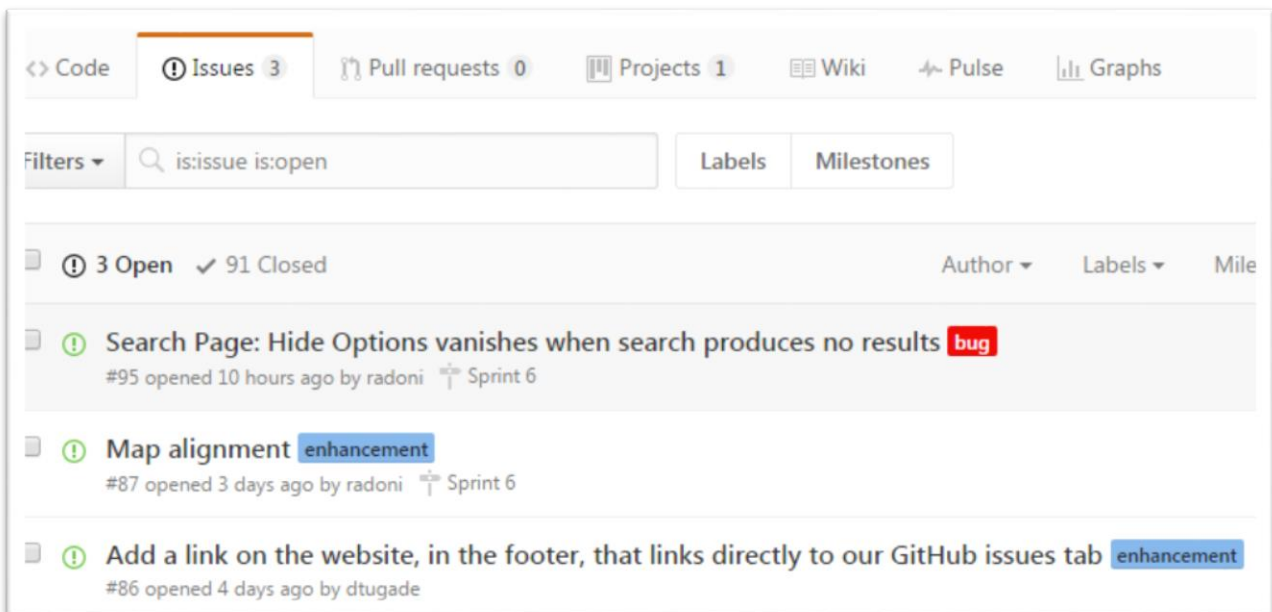
BUGS AND ISSUES MANAGEMENT

Bugs and issues are inherent on any project—particularly a project that involves expedited sprints using open source technologies and development of a new functional prototype using best of the breed tools and frameworks. Because bug and issue management require looking into the future and assessing scenarios that may or may not occur, project teams often do not adequately focus on this activity. Cambria established a rigorous issue identification and monitoring processes early in the project and supported them with formal governance processes. Defects identified early were added to the Issues using GitHub and were addressed either as part of the next sprint or are documented as a product backlog item.

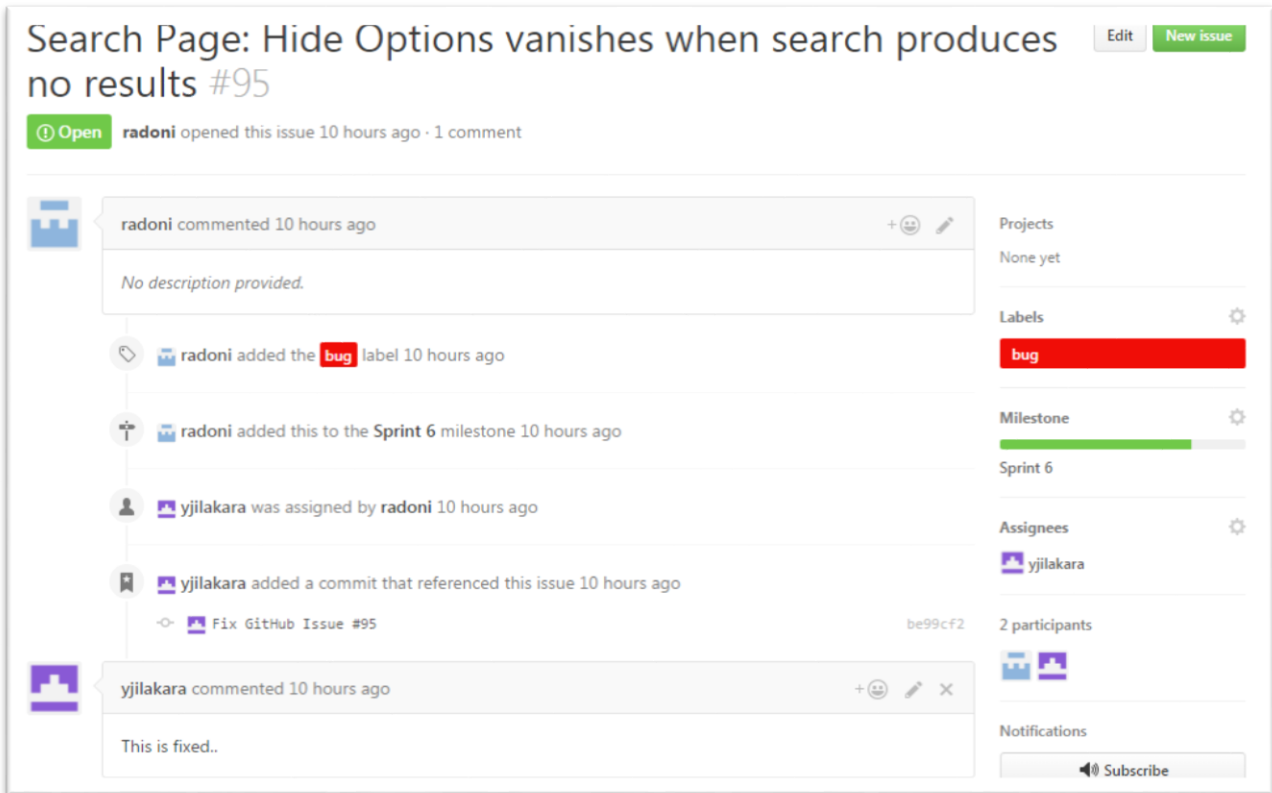
ISSUES FOUND PRIOR TO DEPLOYMENT

Cambria utilized GitHub’s native ‘Issue Management’ module to track and manage issues pertaining to the functionality of the prototype prior to the deployment. The usability testers verified the usability of the application and logged the bugs in a Usability Testing Results document. These bugs were logged into GitHub as Issues and were assigned to the Delivery Manager and the Technical Team. If an issue was deemed an enhancement, the delivery manager worked with the product manager to either 1) include the issue in a sprint backlog, or 2) include the issue(s) in the product backlog.

Here are some key points on how GitHub’s ‘Issue’ module was used by the Cambria team. The typical Issue module would look similar to the screen shot below. It would contain all the issues that are open so the user can see all the issues that are currently open.



The User can also see the issues that are closed by simply choosing the ‘Closed’ option. A typical issue in GitHub looks a bit like this:



- The **title** and **description** described what the issue is all about.
- **Labels** helped the Cambria team categorize and filter the issues (just like labels in email).
- The **milestone** acted like a container for issues. This was useful for associating issues with specific rollout phase – i.e. Sprints.
- Cambria team's operating process included assigning the issue to an **assignee** that was responsible for working on the issue.
- **Comments** allowed anyone with access to the repository to provide feedback.

MILESTONES

Since Cambria team utilized a true Human Centered Design process to design and develop the prototype, the team developed and tested the prototype using six different sprints. The Cambria team used the Milestone feature to assign the issues to a representative sprint.

Set milestone

Filter milestones

Open

Closed

Sprint 5

Closed a day ago

Sprint 4

Closed 4 days ago

Sprint 2

Closed 8 days ago

Sprint 3

Closed 6 days ago

Sprint 1

Closed 8 days ago

Sprint 0

Closed 15 days ago

LABELS

Cambria team used labels to organize different types of issues. Issues were tagged with as many labels as the team wanted, and the users are able to filter using either one or many labels.

Apply labels to this issue

Filter labels

☒ bug

☐ duplicate

☐ enhancement

☐ help wanted

☐ invalid

☐ low priority

☐ question

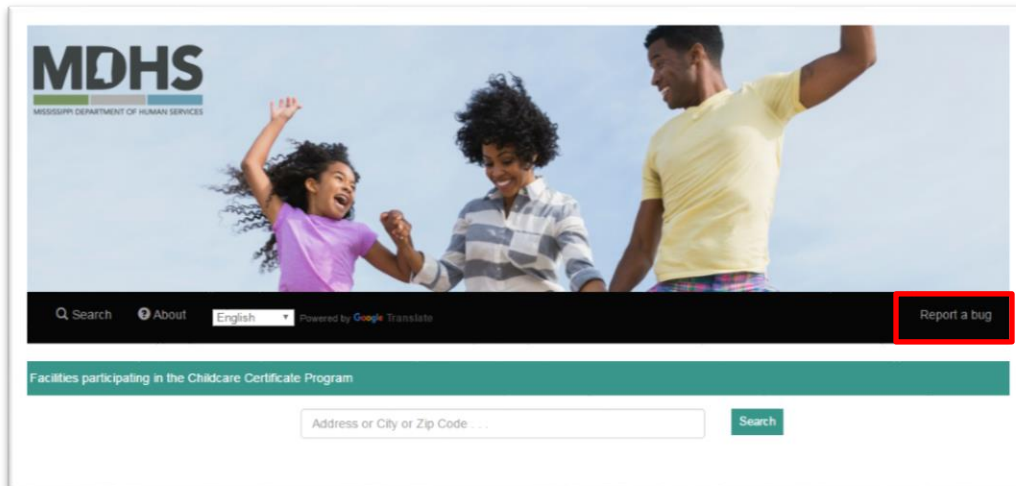
☐ user story

☐ wontfix


☐ Work

ISSUES POST DEPLOYMENT

The users that will be using the prototype may not be having access to GitHub. Due to this potential scenario, Cambria team realized GitHub may not be a plausible solution to report bugs and issue when using the prototype. Therefore, Cambria team implemented a feature in the prototype for users to log and transmit the issues to the development team.



Users, once logged into the prototype can report a bug or an issue by clicking on the ‘Report a Bug’ button on the portlet of the application. The ‘Report a Bug’ feature will enable users to send the issues via an email to Cambria Marketing team. Cambria team will then log and track the issue to closure. The current design of the email is a free form text format – i.e. the user can send any/ all pertinent information related to the issue.

 Send	To...	CSINFO@CAMBRIASOLUTIONS.COM
	Cc...	
	Bcc...	
	Subject	