

Service

Scrumbags final project report

Tony Ciminello

Nick Petty

Raquel Rosa

Gavin Wolf

Contents

About the product	1
About the team.....	2
WebRatio	2
Circuit	3
Trello	3
Git.....	4
Cameo	5
Testing.....	6
Next steps	6

About the product

Service is a web application that allows users to inform property managers of needed maintenance. The primary goal of the product is to streamline and crowdsource the care of public and private property. We hope to reach as wide an audience as possible and have a positive impact on the spaces we all share and inhabit. Simplicity is what separates Service apart from its competitors; anyone can easily make an account then view and create “service requests.” These action items are the backbone of the application; users make them and property managers respond to them. There are no other extraneous distractions to dilute engagement with Service’s core functionality.

Working with limited time and software knowledge, a few basic user stories have been implemented to create the minimum requirements of the Service application. The most important feature available is the service request; users can view and create them, administrators can modify and delete them. The other primary feature Service provides is the ability to create accounts and log into the system. Once an account is created, service requests can be made, and administrators can delete or modify these accounts. Work on adding the

additional role of property manager, with the ability to respond to service requests, is currently underway and represents the last major feature to be implemented. Once this core set of functions is available, Service is ready for general availability.

About the team

Service has been built by the Scrumbags team. Nick Petty is the product owner, and took leadership in the initial proposal, project management via Trello, and testing. Tony Ciminello is the scrum master and provided valuable input on all aspects of development, drove team collaboration, and worked tirelessly on the website's design. Raquel Rosa is responsible for system architecture, having led the creation of both use-case and BPMN diagrams, and took an active role in tools training. Gavin Wolf is our primary developer after completely rebuilding the website, and the best T-shirt maker a team could ever ask for. Over the course of the semester, Scrumbags lost Josh Rodriguez-Santiago, who had been the major contributor to the initial product build.

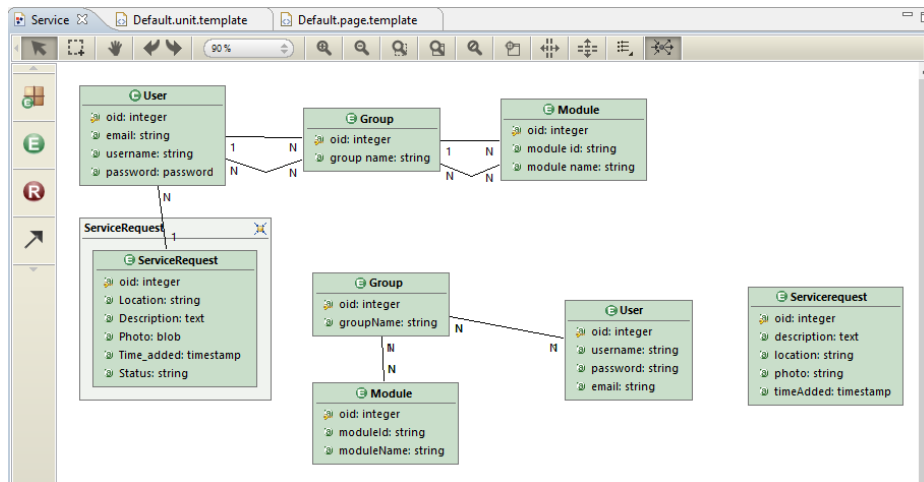
Everyone feels that they did their best to contribute to group success, and that all team members were able to share the workload evenly. The peer evaluation distributes points as follows:

<i>Team member</i>	Tony Ciminello	Nick Petty	Raquel Rosa	Gavin Wolf
<i>Percentage</i>	25	25	25	25

Additionally, because Nick gave presentations on Git, Circuit, and Testing over the course of the semester, the Scrumbags would like extra credit for this project.

WebRatio

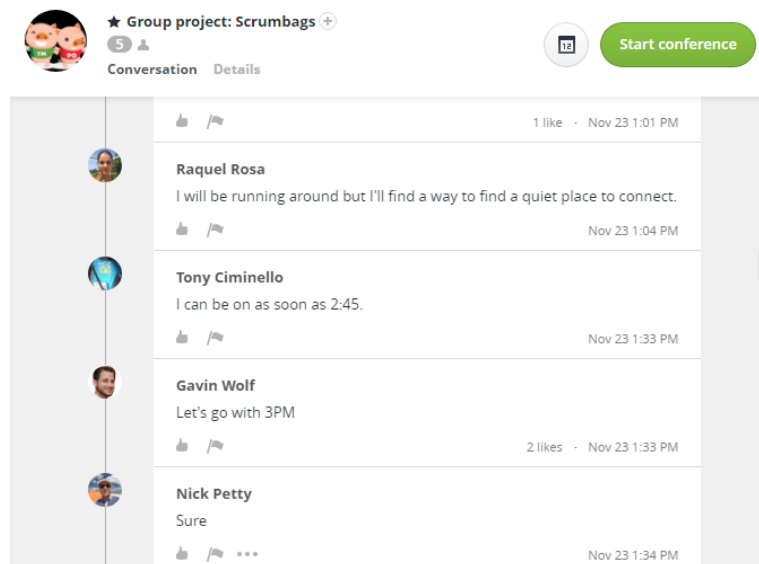
The single greatest obstacle to success in this project has been WebRatio. Multiple team members were unable to open the IDE or deploy their projects consistently, and a single change could take minutes to verify. The problems with this software eventually cost us a team member, as Josh was unable to complete assignments and decided that he would have to drop the class. All development work, including UI design, database management, and coding was done in WebRatio. Balsamiq was considered for website mockups, but WebRatio's difficult-to-modify elements meant design options were limited, and several team members' trial periods had expired. Amazon Web Services was used when WebRatio's database failed, but the website was not stable when using AWS, so this alternative was dropped. Coding was done primarily in WebRatio's IDE, but text editors were used whenever possible. It's possible that WebRatio could be a powerful tool for all-in-one website design and deployment, but the clunky interface, unstable hosting servers, and poor documentation made this project nearly impossible. The Scrumbags do not intend to use WebRatio again in the future.



WebRatio work space, both database and site views

Circuit

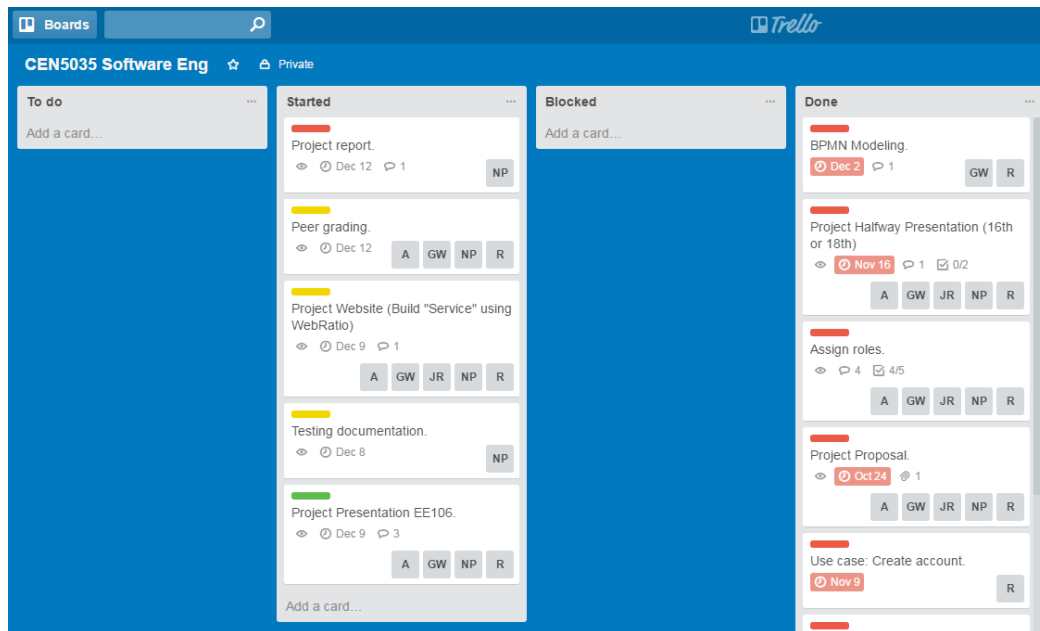
Because Nick is a Unify employee and is driving this class's proof of concept usage of Circuit, the team was very engaged with this application. Once the project milestones were established, weekly meetings were held both in person and through Circuit. Besides this, all team members were active in the team's private conversation and regularly chatted throughout the day. Circuit worked very well in driving collaboration and communication.



The Team's Circuit conversation

Trello

The team used Trello as a Kanban board, as opposed to a Sprint-based backlog. The board is split into four task classifications - to do, in progress, blocked, and done. Tasks are tracked by cards. The cards are prioritized by color labels; red is for high priority, yellow is for medium, and green is for low. Names on cards indicate responsibility for the task, and due dates are included when available. There was debate about whether to use Trello or Jira, but Jira has a steep learning curve and this project is not complicated enough to warrant such a powerful tracking system. Everyone on the team could plan and manage their tasks easily with Trello.



Trello taskboard

Git

File sharing and code management was done through a GitHub repository. Git worked well to provide a centralized, accessible, history-tracking storage system. Unfortunately, not all software used integrated well with Git, or if integrations were available, the team was not knowledgeable enough to make use of them. Specifically, WebRatio and Cameo did not have a known way to natively use Git within their IDEs, or with the source code they used. As such, workspaces were simply zipped and stored on the repository, which created problems with deployment and environment configuration when multiple people attempted to use the same project.

Ninjinkai / ScrambagsProject

Unwatch 4 Star 0 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Project for CEN5035 — Edit

56 commits 1 branch 0 releases 5 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

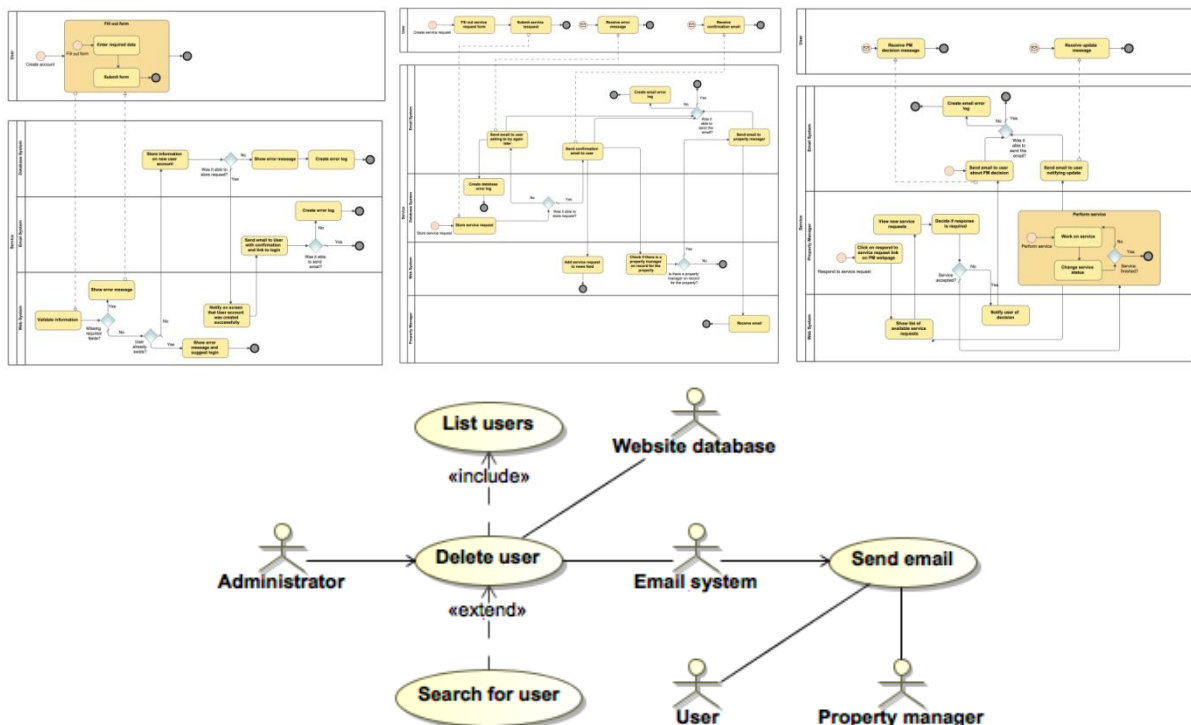
Ninjinkai committed on GitHub Revised final project .docx Latest commit bd87b99 a day ago

File	Commit Message	Time Ago
Delete_user	Use case document and diagram completed.	a month ago
Respond to Request	Update Project with Homework 2	6 days ago
Anthony_Ciminello_Scrambags_use...	Add files via upload	a month ago
CEN5035_Fall2016_P2_Project_Prop...	Requirements posted.	2 months ago
CEN5035_Fall2016_Project Presenta...	Outlined final project report.	6 days ago
CEN5035_Fall2016_ProjectDescripti...	Requirements posted.	2 months ago
Computer Repair BPM Tutorial.pdf	Uploaded BPMN tutorial.	22 days ago
Final Project Report.docx	Added a little more to the report.	2 days ago
Gavin_Wolf_Scrambags_use_case_Cr...	Submitted	a month ago

GitHub view of repository contents

Cameo

UML and BPMN assignments were done in Cameo Enterprise Architecture, much to the chagrin of team members who have already spent time with other diagramming software. Cameo is extremely sluggish to run, and its UI and feature set are far too dense for the needs of this project. Some BPMN was also done in WebRatio, but had to be re-done in Cameo to meet project requirements.



BPMN use-case diagrams

Testing

Basic manual testing is done on a live deployment of the website. Tests are tracked with an Excel worksheet. If the team was working on this project for an extended period of time, test automation could be done with Jenkins managing project building and deployment, then Selenium and Sikuli scripts manipulating the website and viewer operating system. Unit testing has not been implemented at all. The WebRatio IDE may provide JUnit integration, but the provided timeframe and work requirements were not sufficient to implement a test-driven approach to development.

Service Sanity Tests					
Date	Tuesday, December 06, 2016				
Tester	Nick Petty				
Not tested	2	20.00%	<div>Pass all tests</div> <div>Reset sheet</div>		
Passed	7	70.00%			
Failed	1	10.00%			
Test Number	Feature	Steps	Expected Behavior	Result	Notes
1. Log in		1. Click the "Login" button. 2. Enter a valid user name and password.	1. The user is logged into the system. 2. The user is directed to the home page.	Passed	Log in window has "UserNam" instead of "User name."
2. Log out		1. After logging in, click the "Logout" button.	1. The user is logged out of the system. 2. The user is directed to the login page.	Passed	
3. Create account		1. Click the "Register" button. 2. Enter a user name, password, and email address. 3. Check that the	1. The registration form is displayed. 2. The account is created. 3. A confirmation email is received. 4. The new account can log in.	Passed	
4. View Service Requests		1. Log in, click on the "Newsfeed" tab, or click on the "Service" logo.	1. The newsfeed is displayed.	Passed	
5. Create Service Request		1. Click on the "Create Service Request" tab. 2. Enter a location, description, and a photo. 3. Click the "Submit" button.	1. The creation form is shown. 2. The new Service Request appears in the Newsfeed.	Failed	Create service request button is missing.
6. Search		1. Click the "Search" tab.	1. The search form is shown.		

Manual testing document

Next steps

For Service to be a viable product, property manager support systems must be implemented. The property ownership verification process needs either real people or an automated system to provide a way for service requests to be handled. Furthermore, administrators should not be the only people responsible for community management. Posts must be monitored by both the users of Service, and another automated image-recognition system. The problems of verification and community oversight are compounded by the potential scope of Service. Assuming the site scales to even a million users, the task of management is practically impossible without a huge human work force. Being built on WebRatio's platform, it's also abundantly clear that a real product would have to be moved to more reliable hosting. While these steps will not happen, the Scrumbags team is satisfied with Service as it is submitted.