

Neighborhood Social

Bryan Mort

Tim Morris

Jordan Roucco

Lloyd Mageo

Capstone Project [Fall, 2019]

The University of West Florida

12/6/2019

CIS 4595

Dr. Bernd Owsnicki-Klewe

Abstract

With the project coming to a close, it is important to reflect the work that was done. As a team, we are very proud of our accomplishments of this project. We began not entirely being sure how to build a system like this, and in the end, we were able to accomplish almost everything we set out to. These accomplishments were only possible by the use of standards created in the beginning and doing our best to follow the path we originally set out as a team.

Table of Contents

Table of Contents	
Abstract	2
Table of Contents	3
List of Figures	4
List of Tables	5
1 Executive Summary	6
2 Final Requirements	7
3 Final Timeline	7
4 Project Results	9
5 Project Process Review	9
6 Work To Be Done	11
Appendix A: Glossary	13

List of Figures

Burnup Charts:

- | | | |
|----|-----------------|----|
| 1. | Sprint 1 Burnup | 10 |
| 2. | Sprint 2 Burnup | 10 |
| 3. | Sprint 3 Burnup | 11 |

List of Tables

Project Timeline tables:

1.	Original Timeline	8
2.	Final Timeline	8

1 Executive Summary

Neighborhood Social aims to connect neighbors with one another through an exclusive social web application. Neighbors are able to create posts of various motives including for sale items, upcoming neighborhood events, and services. Each member must be able to prove their credentials in order to securely connect to their community's platform.

2 Final Requirements

Our requirements from the initial plan did not change much from our final product. The only requirement that did change was our deployment platform. For our database, we used MongoDB with a back-end API built using Java as stated in our project plan. We used HTML and CSS for our front-end GUI which was also stated in the project plan. As for the deployment platform, we had to change from AWS ElasticBeanstalk to a laptop Ubuntu server that was stored in our Server Admin's guest room that runs an instance of Apache Tomcat for deployment . This was because ElasticBeanstalk has issues connecting with Mongo databases. After encountering this issue, we quickly switched gears and created a new solution that would allow for an effective product.

3 Final Timeline

As can be seen by the timeline tables, our timelines changed a bit. Originally, we planned on four sprints of three weeks with one sprint being two weeks, however, we decided to switch it to three sprints of four weeks each. This would give us more time to focus on completing boards stated in our product backlog while having the group members not feel rushed to complete their work. Since time management can be very difficult with all members having several classes, it was important that we consider that the shorter sprints may restrict our development process.

Project Plan Timeline:

Sprint	What to Accomplish
1 (Weeks 1 - 3)	<ul style="list-style-type: none">• Database creation• Home page developed• Front to back communication
2 (Weeks 4 - 7)	<ul style="list-style-type: none">• Neighbor creation• Credentials page implemented• Post creation• Front page developed
3 (Weeks 7 - 9)	<ul style="list-style-type: none">• Front page filtering and sorting options implemented• Reward system
4 (Weeks 10 - 11)	<ul style="list-style-type: none">• Final testing

Final Timeline:

Sprint	Accomplished
1 (Weeks 1 - 4)	<ul style="list-style-type: none">• Database creation
2 (Weeks 5 - 8)	<ul style="list-style-type: none">• Front to back communication• Neighbor creation• Credentials page implemented• Home page developed• Front page developed (barebones)• Post creation• Front page filtering and sorting options implemented
3 (Weeks 9 - 11)	<ul style="list-style-type: none">• Front page developed (design)

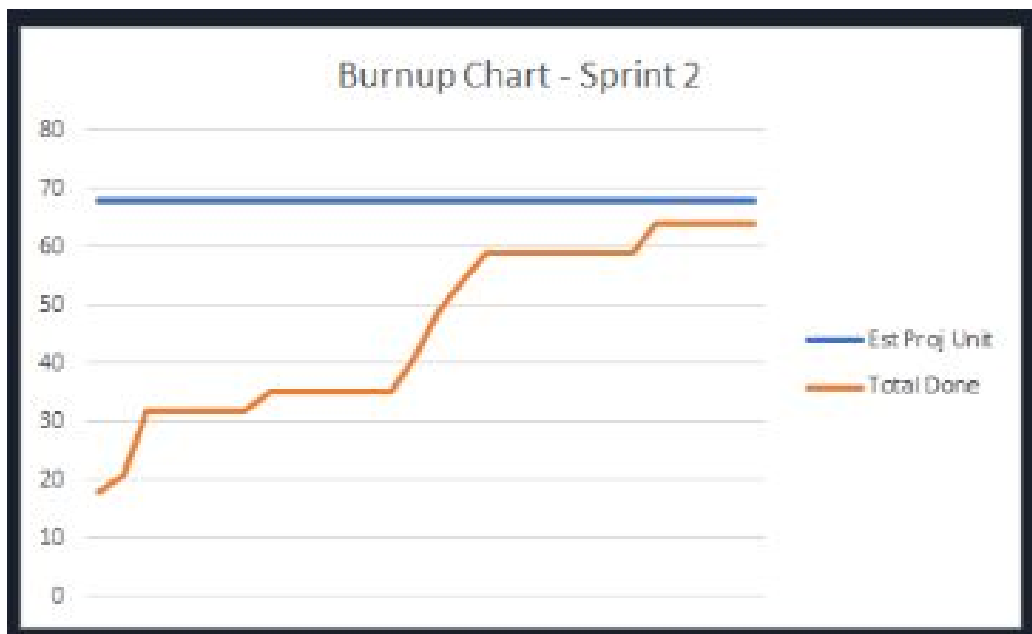
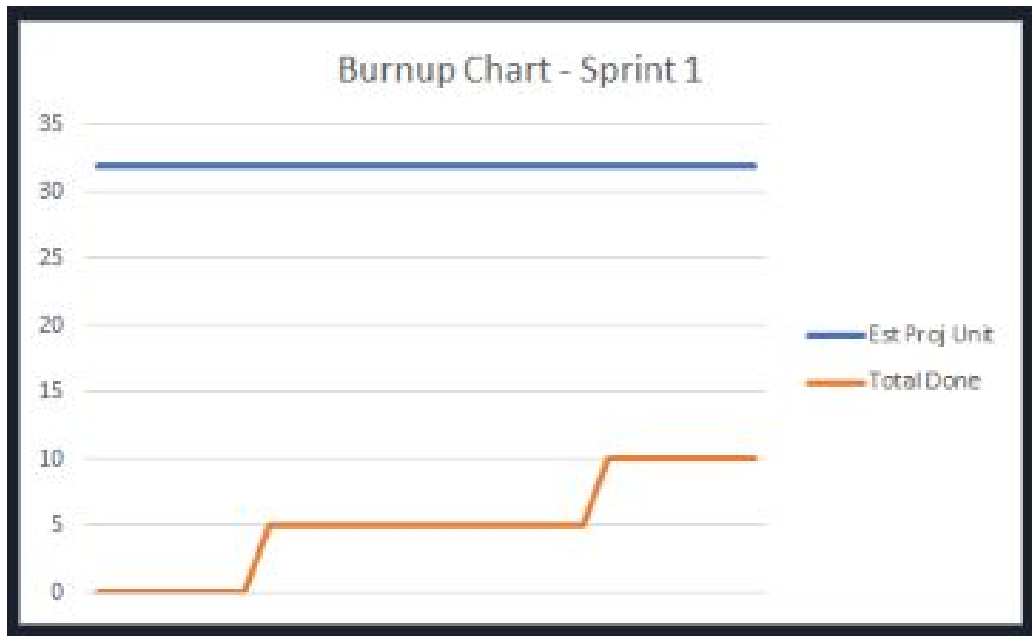
4 Project Results

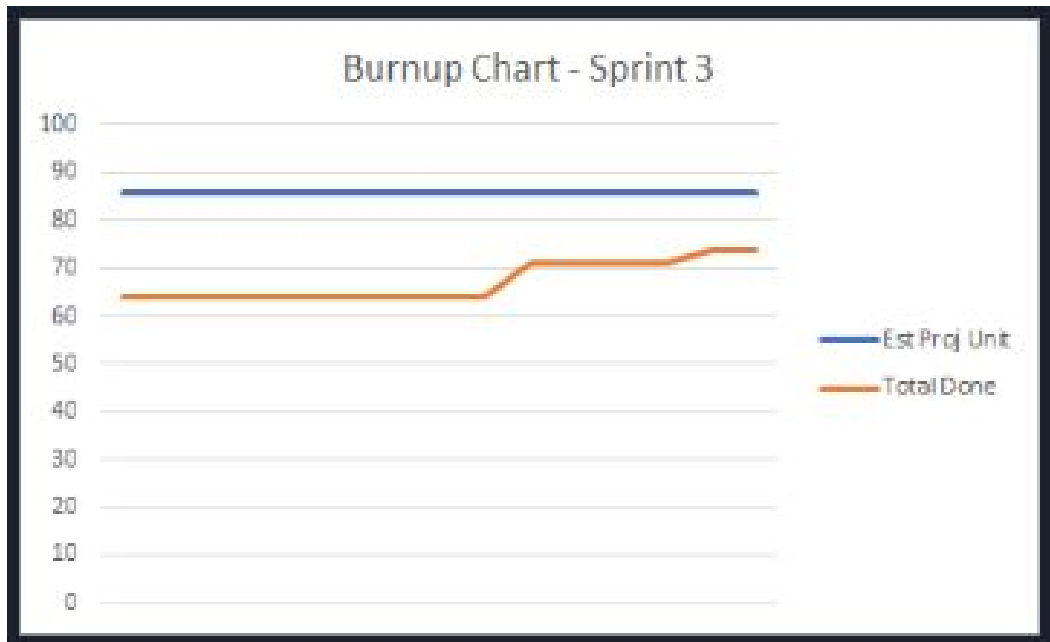
After three months of development, we are proud of our resulting application. Users are able to create their profile, log in with their credentials, view posts on their homepage with sorting and filtering options, create posts to share with their neighbors, and able view a list of their neighbors along with the ability to contact each neighbor.

5 Project Process Review

As stated in the project plan, we used an agile process within the SCRUM framework. This involved creating user stories within Trello that we used to create a product backlog. We then sorted the stories out into sprints that we used for sprint planning and eventually creating a sprint backlog. During this sprint backlog initiating, we removed some stories we believed we would not be able to get to, however, some of these stories naturally came to fruition through developing the application. Future uses of Trello need to include these additional developed functionalities.

At the end of each sprint, a burnup chart was created. This allowed for the SCRUM master to view the amount of progress that was being done by the group for the overall sprint. After analyzing the first sprint, the burnup chart showed that little progress was made. Although it may have looked like nothing was getting done, we actually spent the majority of that sprint figuring out how the different systems would work, which included the change in deployment platforms. After working out the kinks, we were able to make immense progress as is shown in the sprint 2 burnup chart. With the third burnup chart - which was cut short a few weeks because of finals and project turnins - we can see that we came very close to fulfilling all of our story points that were decided upon during the product backlog creation.





6 Work To Be Done

Although we are very proud of our final product, there are a few issues we wish we had time to address. Unfortunately, the GUI of a few of the pages is not fully developed. Given more time, we would be able to finish creating the website we had originally envisioned. Another would be a notification of new posts. This would simply be a flag over a post type that would alert the user that there are new posts of that type. Another would be an invite system that would have also included a Local Agent user to the system. A user with certain credentials would be able to invite new users with invitation codes. Creation of this user is paramount in the final development of the application. With security being a top priority of this application, we wish we were able to include a secure storage and retrieval of service account credentials, which would allow certain functions to only exist within an account login. This would prevent processes from accessing information that is not necessary for the job being done on the system.

Appendix A: Glossary

Term	Definition
GUI	Graphical User Interface
API	Application Programming Interface
CSS	Cascading Style Sheets
AWS	Amazon Web Services
HTML	Hypertext Markup Language
Elastic Beanstalk	An AWS service that is used to host the website. Provides the url necessary to access the website as well as related resources.