Feb 2021

# Crowd Canvass
# Project Management Verification Assignment

**COSC 4920-102**
**Crowd Canvass**
**Shayne Burns**
**Ramzi Carter**
**Wylie Frydrychowicz**
**Max Rothweiler**
**Hannibal Santiago**
**Submission Date: 02/03/21**

**Table of Contents**

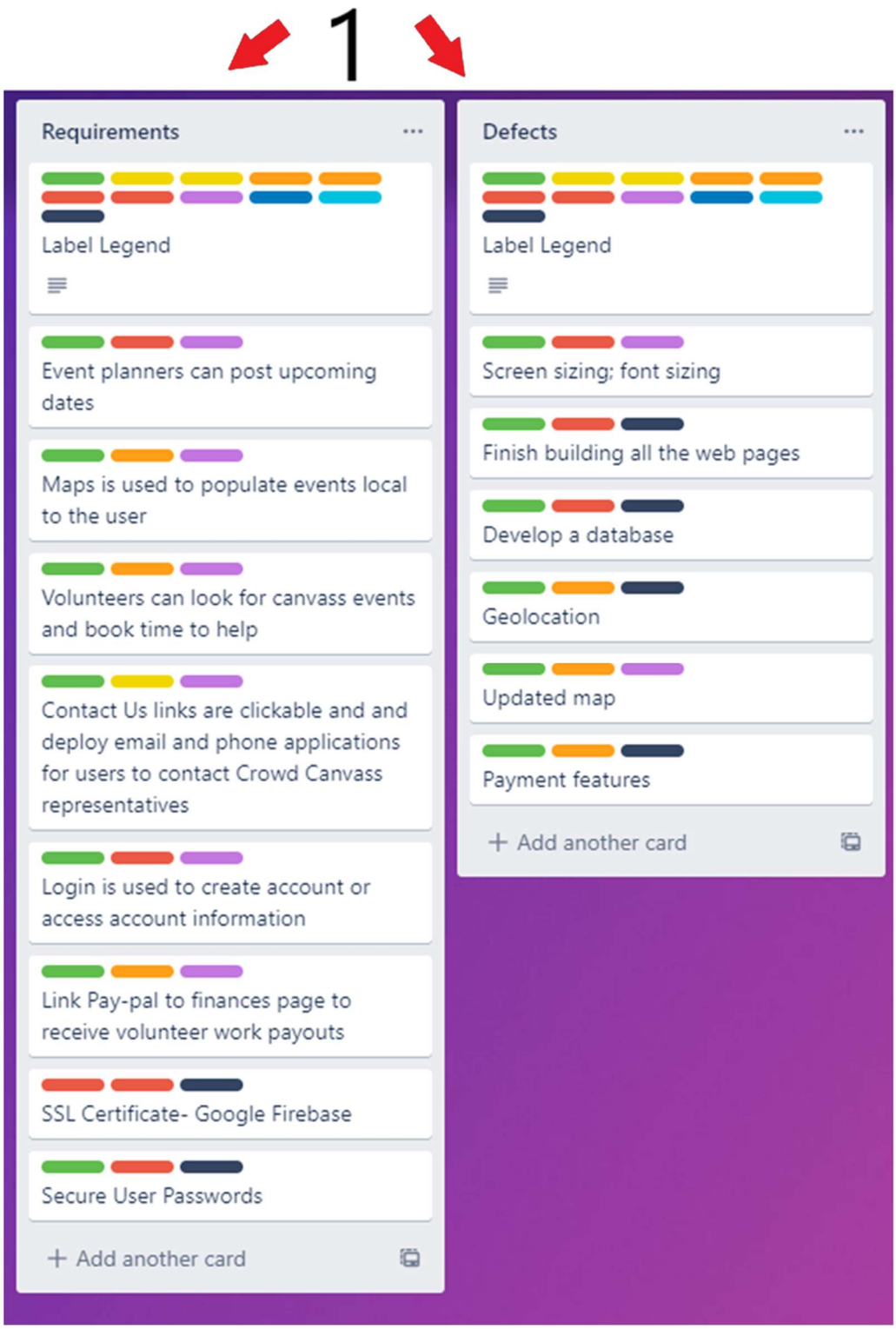


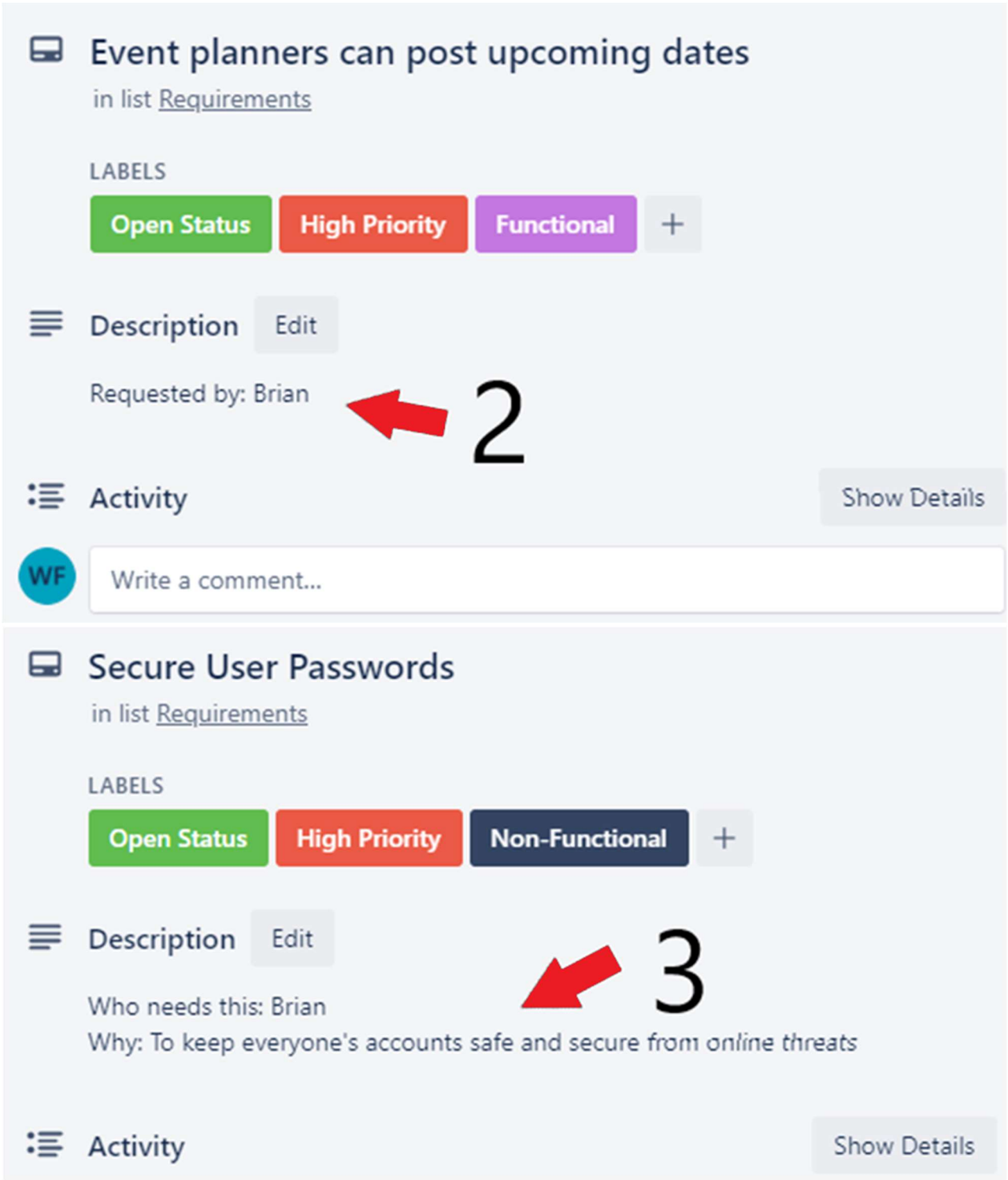
# 2. Project Management Tool Selection

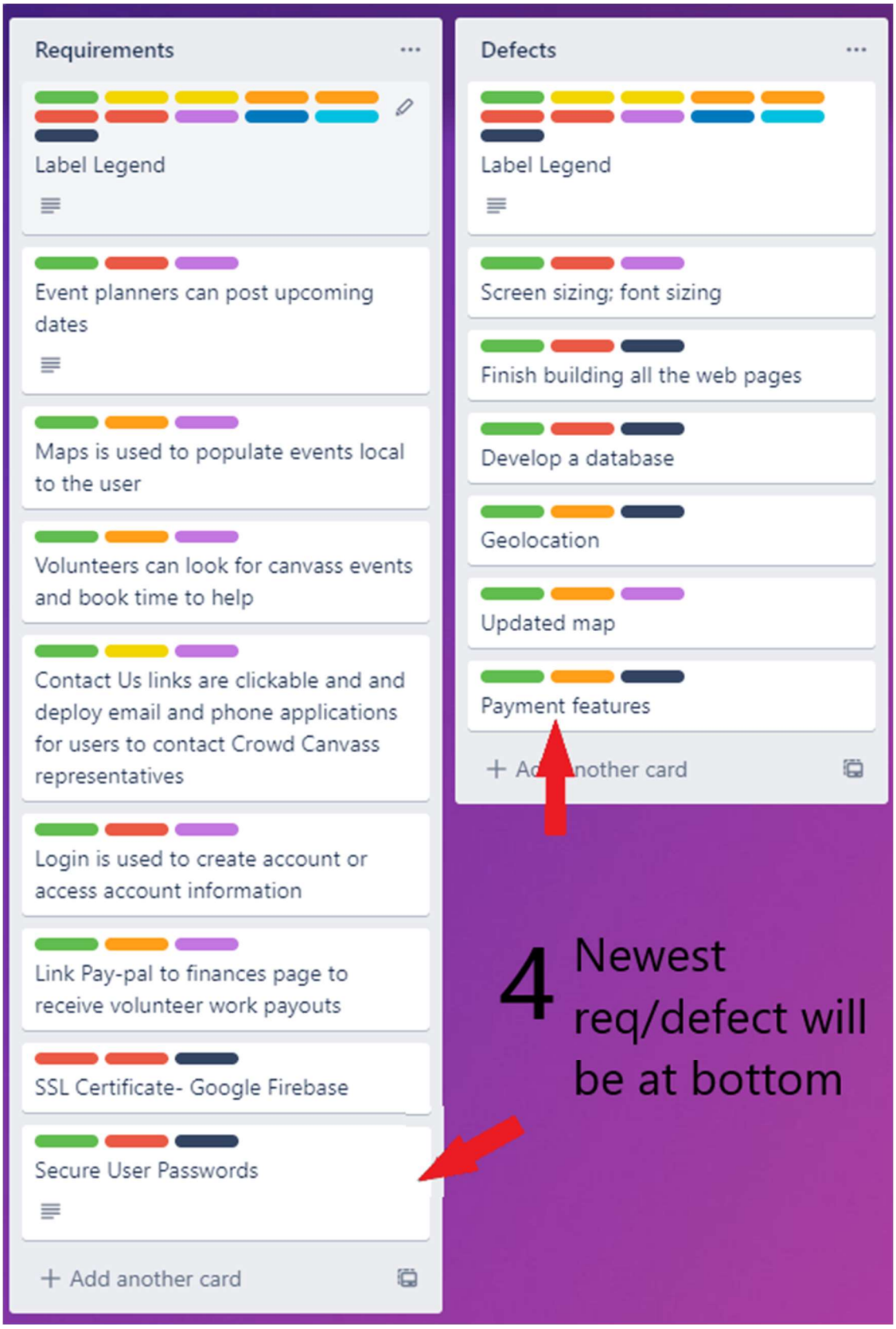
The project management tool our group chose was Trello. When determining what tool we would select, the main criteria we looked for was a web-based application that would enable us to create categories used to organize and plan the various aspects of our project. It was also important that the tool would be easily accessible and editable by all team members. Trello encompassed these criteria and is known to be an effective project management tool many in the industry. The link to our Trello for Crowd Canvass is included below.
https://trello.com/b/IAMrTxNv/crowdcanvass

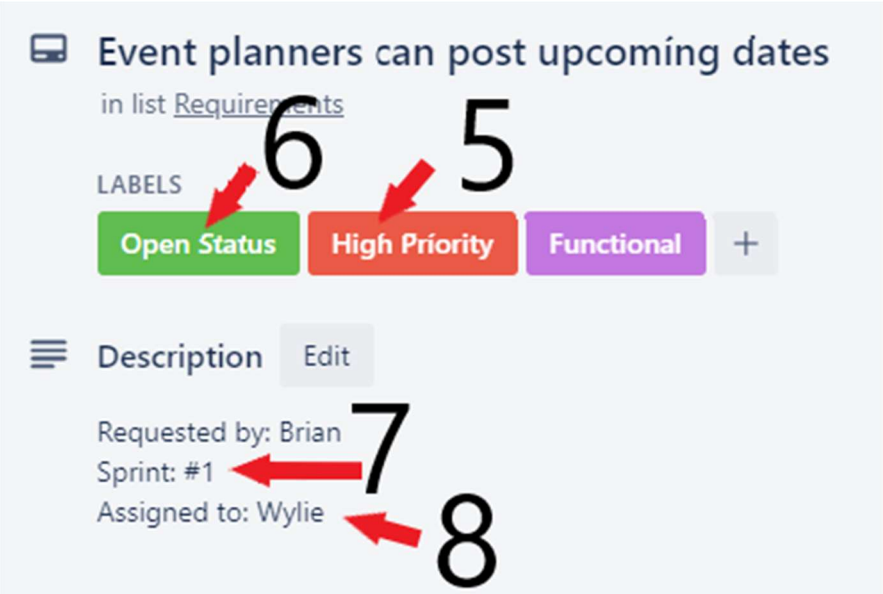
# 3. Demonstration of Project Management Tool:

## 3.1 Project Management Tool Implementation/Configuration:




Event planners can post upcoming dates
in list Requirements
LABELS
Open Status
High Priority
Functional
Description
Edit
Requested by: Brian
2
Activity
Show Details
WF
Write a comment...
Secure User Passwords
in list Requirements
LABELS
Open Status
High Priority
Non-Functional
Description
Edit
3
Who needs this: Brian
Why: To keep everyone's accounts safe and secure from online threats
Activity
Show Details


Requirements
Label Legend
Event planners can post upcoming dates
Maps is used to populate events local to the user
Volunteers can look for canvass events and book time to help
Contact Us links are clickable and and deploy email and phone applications for users to contact Crowd Canvass representatives
Login is used to create account or access account information
Link Pay-pal to finances page to receive volunteer work payouts
SSL Certificate- Google Firebase
Secure User Passwords
Add another card
Defects
Label Legend
Screen sizing; font sizing
Finish building all the web pages
Develop a database
Geolocation
Updated map
Payment features
nother card
4
Newest req/defect will be at bottom




1.) A list of requirements/defects, demonstrating that there is, in fact, a list, not just a one or two entries.
2.) Identify a Functional (user) requirement/defect. Identify where the "user(s)" is indicated. This can also be an Actor name. (i.e., who requested this?) The user can be listed in the subject, in a separate field (better), or in the text, just so long as it is consistent.
3.) Identify a Non-functional requirement. Show where the stakeholder(s) is indicated. (i.e., who needs this?) The user can be listed in the subject, in a separate field (better), or in the text, just so long as it is consistent.
4.) Identify how to determine/indicate a new feature/requirement v. a bug/defect
5.) Identify the Priority of a requirement/defect (often high/medium/low)
6.) Identify the Status of requirement/defect (often open/in progress/in test/closed/duplicate/rejected/etc.)
7.) Identify the sprint to which this requirement/bug is assigned (a sprint can be identified by a date or a revision number)
8.) Identify which team member this requirement/defect is assigned to

## 3.2 Requirement/Defect Template:

# CROWD CANVASS

Crowdcanvass@gmail.com · (414) - 585 - 1200

---

Scrum Meeting Requirements and Defects Report

## REQUIREMENTS

**FUNCTIONAL**

- Event planners can post dates for upcoming volunteer opportunities
- Integrated map that populates canvass events local to your area or in desired search ranges
- Volunteers registered to Crowd Canvass can look for events and book time to help
- Users can create a username and password for login and access personal account information
- Connect payment information to accounts page via Pay Pal for payout or charity accommodations from volunteering parties
- Contact Us links and tabs are clickable and direct personnel to next path on website

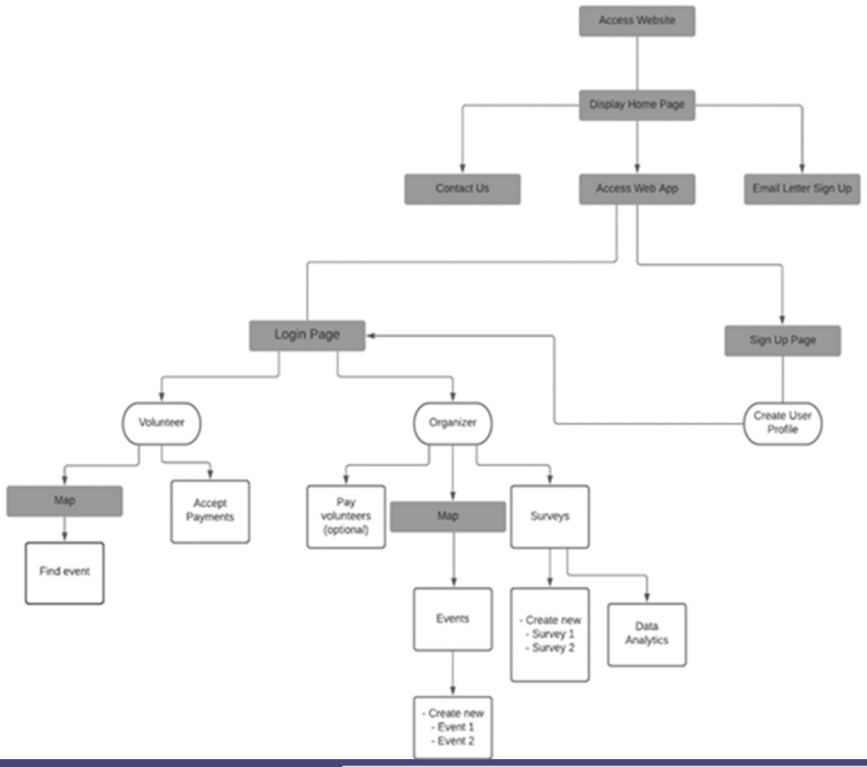
**NON-FUNCTIONAL**

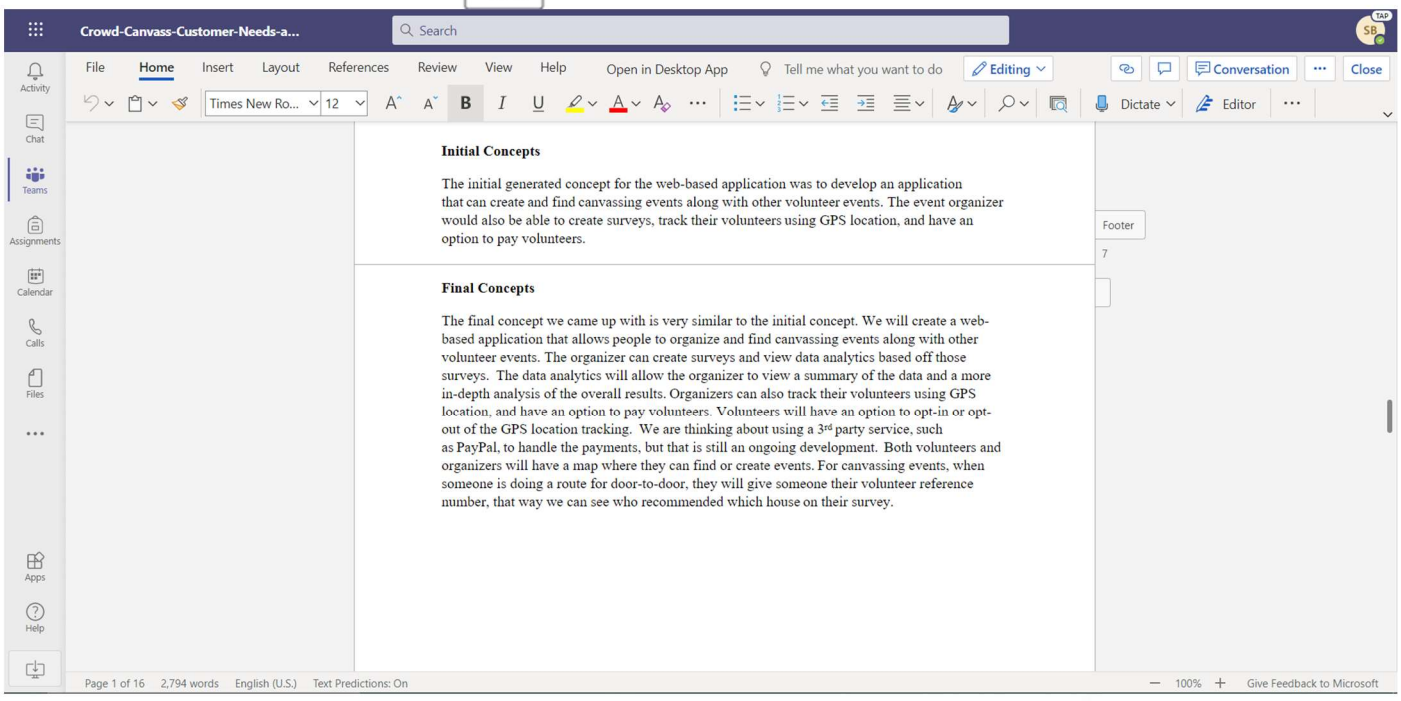
- SSL Certificate via Google Firebase
- Security provision for user account information via hash sets and encryption
- Usability improvements for design, colors and navigation management
- Implement storage for individuals creating accounts and use external cloud computing for a high volume of users, which will improve performance for website when experiencing many active users
- Working links direct user to next destination in timely manner

## DEFECTS

- Login functionality is not working properly, more code needs to be implemented
- Font size and text format does not display accurately
- Some links are not fully functional
- Page spacing and screen sizing is not optimized for mobile computing as desired

# 3.3 Reference from Prior Notes Workflows:




Our requirements, defects, and user stories follow a simple flow found in the screen shots above. If you were to follow the creating events feature with GPS you start with our initial concept then the workflow. We then break down how we want our applications to work. Then from their meetings to refine our features especially the one with our Stakeholder Brian. After that the creating events and the GPS features are put into a final concept. These pieces are then broken down in the Requirements and user stories that can be viewed on out project management tool.

# 4. Testing Approach:

*The way to test our system would be to first do unit testing with isolating a list of services or features such as: GPS tracking, the map updating, the filter system working with both listed events and the map, account set up and management, and making sure either bank accounts are linked to accounts or PayPal. These features, or services will be tested together afterwards in the System testing practice, and this would be used particularly to functionality across different services or features, such that they work together, and that the security of the account information and payment features are secured. We would also make attempts to hack into our own system, to ensure the security of the accounts and their information. Following this, is the Program testing, this is where we would work out all the bugs, as well as errors, and we would ensure that the fluidity of the platform is established and preserved. We would test for the bugs by establishing test accounts and testing sites to track the payment features, GPS features, as well as the filtering service. Our method of collecting and holding data is seen by the template below:*

| Unit Testing/System Testing/Program Testing | Is it compiling? | Is the service/feature working when attempting to test independently? | Is the device working cooperatively? |
|---|---|---|---|
| GPS tracking | | | |
| Map updating | | | |
| Filtering System | | | |
| Security | | | |
| Encryption | | | |
| Payment Linkage | | | |
| Collective Data | Does this work for 1 account? | Does this work for 50 accounts? | Does this work for 500 accounts? |
| Tracking of the account | | | |
| Can an account send payments | | | |
| Can an account receive payments | | | |
| Does the filtering by Zip Code/Geolocation work? | | | |

*Figure 1: This is the template used for testing and data collecting*