Cheap Eats

Sprint Review Report
Sprint 1
Version 1
June 25, 2019

Nathan R. Hall
Software Engineer
Christian Ford
Software Engineer
Benjamin Miller
Software Engineer
Scott Sheffer
Software Engineer

Prepared for
CS1530-1010
Summer 2019
University of Pittsburgh

Revision History

Date	Description	Author	Comments
6/23/19	Version 1	BM, SS, CF, NH	Start filling out some of the stories
6/25/19	Version 2	BM, SS, CF, NH	Complete filling out the SR1

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
Benjamin Miller	Benjamin Miller	Software Eng.	6/25/2019
Scott Sheffer	Scott Sheffer	Software Eng.	6/25/2019
Christian Ford	Christian Ford	Software Eng.	6/25/2019
Nathan R. Hall	Nathan R. Hall	Software Eng.	6/25/2019

Table of Contents

1. Introduction

The goals of Sprint 1 are to create a documentation of the stories that must be completed for Cheap Eats to be functional. The stories must be identified, and analyzed to better understand how difficult each will be to complete, and how each story interacts with the other stories. Cheap Eats will allow Users to create an account, register it, and verify it with a university email address. Additionally, Users will be able to make, like, dislike, rate, and flag Posts, as well as block specific Hosts that they would not like to see any events from. As Users browse the application, they will be able to set their notification settings and their food preferences, as well as favorite specific Hosts that they would like to keep up-to-date information on their events. Users can filter their browsing experience by a multitude of filters to ensure they only see the kinds of events they want to see.

This project has 31 stories, and 132 points. 0 points were completed thus far. The overall project direction is slow. One major challenge to the success of this project is the difficulty in coordinating schedules to allow for the development team to meet with regularity. Another challenge has been the initial adoption of Android Studio and setting up the environment necessary to productively work on the Cheap Eats Application.

2. Specific Goals

2.1 1 ACCOUNT - Verify University Email on Registration

2.1.1 Story Description:

- Upon creation of a User Account, the User will be prompted to register their account, and upon registration the User will be prompted to verify their account using their University email address.

2.1.2 Story Acceptance Criterion

- Verification Email has been sent to the user with a confirmation link that the user needs to click on in order to verify their account. Once the verification link has been successfully clicked the acceptance Criterion is met

2.1.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- User Login

2.1.4 Story Challenges

- Some challenges faced in this story will be setting up the application to automatically send a verification email upon request. Also checking to make sure that verification email is being sent to a .edu (or another type of education account) is valid.

- Some users may also have 2 separate emails that we will have to handle in this situation. For example, they may sign in and create their account under a google email. But will still need to register their .edu email in order to have full access to the applications functionality.

2.1.5 Story Assigned to

- TBD

2.1.6 Story Points

- 2 Points

2.1.7 Status: Completed or not

- Not Completed. Plan of Action: to being implementation of this along with the other account functionality.

2.2 2 ACCOUNT - Account Creation

2.2.1 Story Description:

- In order to continue using the application a user must first create an account, and register it. This is where the story account creation comes into play. We will implement Account creation with the help of firebase.

2.2.2 Story Acceptance Criterion

- The Account creation is dependent upon reaching the firebase and being able to create an account using a valid email and password to store the user's new account information in the firebase. Once implementation is setup correctly for a user to enter their email & pass, and it's passed and saved into firebase then we can accept the story.

2.2.3 Story Dependencies

- VIEW - User Registration

2.2.4 Story Challenges

- The challenge with this story will be establishing a connection with the firebase to store the new users information. Another challenge will be prompting the user to register in a seamless way that won't cause headaches for the user. (Although this is more under the registration view)

2.2.5 Story Assigned to

- TBD

2.2.6 Story Points

- 6 Points (3 hours of work)

2.2.7 Status: Completed or not

- Not Completed. Plan of Action: to do implementation of this along with the other account functionality.

2.3 3 ACCOUNT - User Login

2.3.1 Story Description:

- Login functionality for the user to have access to every time they open the application. The application will use a one-time login per device using firebase. But anytime the user logs out they will be brought back to attempt a login.

2.3.2 Story Acceptance Criterion

- When the application is capable of taking the email and password parameters and passing it through Firebase and returning a valid or invalid login the story acceptance criteria will be met

2.3.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation
- VIEW- User Login

2.3.4 Story Challenges

- Some difficulties that could come with implementing the User login story would be establishing a valid back and forth from the firebase, also having the user login story work directly with the login view.

2.3.5 Story Assigned to

- TBD

2.3.6 Story Points

- 1 Point (1 half hour of work)

2.3.7 Status: Completed or not

- Not Completed. Plan of Action: to do implementation of this along with the other account functionality.

2.4 4 ACCOUNT - Set Food Preferences

2.4.1 Story Description:

- Set food preferences will work directly with the user in the application to help the user see the foods they prefer more often in deals than the foods they don't prefer

2.4.2 Story Acceptance Criterion

- The acceptance criterion for setting food preferences will be when the user is able to set food preferences and allow it to factor in the clout of posts that they see. Once we can implement so that the set food will act as a manager to the table of events going on for each specific user then we can say that the story will be accepted.

2.4.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation
- ACCOUNT- Update Notification Preferences

2.4.4 Story Challenges

- The main challenge for this will be calculating the weight of preferences vs. the weight of all the other posts, and figuring out how to return it to the user in a way where they can see their preferred foods first but still see other deals.

2.4.5 Story Assigned to

- TBD

2.4.6 Story Points

- 4 Points (2 hours of work)

2.4.7 Status: Completed or not

- Not Completed. Plan of action: implement once the post handler can return a table to a specific user. Then once that is good, we can use this story to filter and reweight the posts.

2.5 5 ACCOUNT - Add / Remove Favorite Host

2.5.1 Story Description:

- Add / Remove favorite is a main part of the account functionality. It is included in this because it allows the user to set Favorite Restaurants / Events that they like to go to. By doing this it allows them to receive notifications immediately about favorites

2.5.2 Story Acceptance Criterion

- The story will be accepted when the back end user manager is able to filter the event list to move “favorited” events to the top. Also when the user manager is able to successfully remove the favorite and stop updating the list when the user calls to “remove” said favorite.

2.5.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation
- POST- Post Access Control List (ACL)

2.5.4 Story Challenges

- Some challenges with this will be handling the accessors to the list of events that the user will receive. With doing so we want to make sure that the user receives their favorites first. Along with this we need to be able to update the list whenever the user decides to remove one of their favorites as well.

2.5.5 Story Assigned to

- TBD

2.5.6 Story Points

- 3 Points (1 and 1 half hours of work)

2.5.7 Status: Completed or not

- Not completed: Plan of action is to complete once the user login and main account creation functionality is done. Also when the post handler is mainly done too.

2.6 6 ACCOUNT/BROWSE/POST - Block a Host

2.6.1 Story Description:

- When a user doesn’t wish to see what a host has to post or just wants to stop seeing anything from them altogether they have the option to Block a host

2.6.2 Story Acceptance Criterion

- Block a host story will be accepted when the implementation is able to successfully allow a user to block a host and the post manager for the user can filter out every single thing about the host. So that the user doesn't have to see anything from them.

2.6.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation

2.6.4 Story Challenges

- Some difficulties that can arise with implementing this will be having the post manager for the specific user refresh on command and filter out everything from that host. Also we have to consider how deep we will allow the block to go, and if we should filter out other posts that mention that said blocked host or not.

2.6.5 Story Assigned to

- TBD

2.6.6 Story Points

- 4 Points (2 hours of work)

2.6.7 Status: Completed or not

- Not completed: Plan of Action: to complete once the Post list can be made and filtered, along with all the other account functionality getting finished.

2.7 7 ACCOUNT - Update Notification Preferences

2.7.1 Story Description:

- Allows the user to update the notification preferences that they will receive about food events and deals going on.

2.7.2 Story Acceptance Criterion

- This story will be accepted when the implementation is able to successfully update the notification preferences for every user and set a specific notification setting for each user. When the implementation is able to do this then we can accept the story criterion

2.7.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation
- POST- Post Access Control List (ACL)

2.7.4 Story Challenges

- Some challenges that we will face with the technical side will be setting the notification for each user separately, when they call upon the update. We may also need to wait for other implementations of the application to be finished before we can start implementing this.

2.7.5 Story Assigned to

- TBD

2.7.6 Story Points

- 8 Points (4 hours of work)

2.7.7 Status: Completed or not

- Not Completed. Plan of Action: implement once all account login / setup functionality is implemented.

2.8 8 POST - Access Control List

2.8.1 Story Description:

- Each Post shall have an Access Control List (ACL) that will be used to enforce proper read and write permissions.

2.8.2 Story Acceptance Criterion

- The ACL only accepts Event Posting instructions from the User making the Post. Additionally, the ACL will allow other Users to Like, Dislike, Flag, and Rate the Post

2.8.3 Story Dependencies

- ACCOUNT - User Creation
- POST - Create Post
- POST - Delete Post
- POST - Edit a Post
- POST - Like a Post
- POST - Dislike a Post
- POST - Flag a Post
- POST - Favorite Post
- POST - Rate Post

2.8.4 Story Challenges

- Ensuring that only specific Users are able to affect Posts in specific ways. I.E. Hosts, should be able to Create, Delete, and Edit their Posts, while other Users can Like, Dislike, Flag, and Favorite Posts

2.8.5 Story Assigned to

- TBD

2.8.6 Story Points

- 2 points (1 hour of work)

2.8.7 Status: Completed or not

- Not Completed. Plan of Action: to complete concurrently with other POST stories, and to use Firebase to aid in the implementation

2.9 9 POST - Create Post

2.9.1 Story Description:

- As a user I want to be able to create a post so that university affiliated events can receive more notoriety and higher attendance.

2.9.2 Story Acceptance Criterion

- Must create an account filling out the mandatory restaurant information (Name, Address, Hours, and Phone Number)
- The account information must be properly stored in the database

2.9.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation
- POST- Post Access Control List (ACL)

2.9.4 Story Challenges

- Some challenges with creating a post will be delegating where the post should go and adding it into the event list, creating a new object for each post with the attributes for it.

2.9.5 Story Assigned to

- TBD

2.9.6 Story Points

- 4 Points (2 hours of work)

2.9.7 Status: Completed or not

- Not Completed, Plan of Action: to work on it concurrently with the other post functionalities.

2.10 10 POST - Edit Post

2.10.1 Story Description:

- As a restaurant owner/manager I want to be able to edit my post so that any changes in event scheduling, restaurant scheduling, or general errors can be attended to properly.

2.10.2 Story Acceptance Criterion

- Must create an account filling out the mandatory restaurant information (Name, Address, Hours, and Phone Number)
- The account information must be properly stored in the database

2.10.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation
- POST- Post Access Control List (ACL)

2.10.4 Story Challenges

- Some challenges that will arise with editing a post will be giving power to edit only the user who created the post. It will also bode a challenge of recreating the post into the post list with the same clout attributes so that it doesn't change clout once edited.

2.10.5 Story Assigned to

- TBD

2.10.6 Story Points

- 4 points (2 hours of work)

2.10.7 Status: Completed or not

- Not completed, Plan of Action: to complete concurrently with all the other posting functionalities, more specifically after Create a Post is done. That way a created post can be edited.

2.11 11 POST - Like Post

2.11.1 Story Description:

- As a user I would want to be able to like a post so that if I enjoy an event I can show support with a like.

2.11.2 Story Acceptance Criterion

- Must create an account filling out the mandatory regular user information (Username, Email, Avatar, Food Preference, Notification Preference)
- The account information must be properly stored in the database

2.11.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation
- POST- Post Access Control List (ACL)

2.11.4 Story Challenges

- Keeping an up-to-date tally of the number of Likes a Post is receiving, and displaying an accurate count to Users.

2.11.5 Story Assigned to

- TBD

2.11.6 Story Points

- 1 Point (1 half hour of work)

2.11.7 Status: Completed or not

- Not Completed. Plan of Action: to complete this story concurrently with the other POST stories

2.12 12 POST - Dislike Post

2.12.1 Story Description:

- As a user I would want to be able to dislike a post so that if I do not enjoy an event I can show the Host with a dislike.

2.12.2 Story Acceptance Criterion

- Must create an account filling out the mandatory regular user information (Username, Email, Avatar, Food Preference, Notification Preference)
- The account information must be properly stored in the database

2.12.3 Story Dependencies

- ACCOUNT- User Registration
- ACCOUNT- Account Creation
- POST- Post Access Control List (ACL)

2.12.4 Story Challenges

- Keeping an up-to-date tally of the number of Dislikes a Post is receiving, and displaying an accurate count to Users.

2.10.5 Story Assigned to

- TBD

2.12.6 Story Points

- 1 Point (1 half hour of work)

2.12.7 Status: Completed or not

- Not Completed. Plan of Action: to complete this story concurrently with the other POST stories

2.13 13 POST - Flag Post

2.13.1 Story Description:

- As a user I want to flag a post so that if inappropriate content or incorrect information is discerned it can be recognized by the moderation system.

2.13.2 Story Acceptance Criterion

- Must create an account filling out the mandatory regular user information (Username, Email, Avatar, Food Preference, Notification Preference)
- The account information must be properly stored in the database

2.13.3 Story Dependencies

- ACCOUNT - User Registration
- ACCOUNT - Account Creation
- POST - Access Control List (ACL)

2.13.4 Story Challenges

- Keeping an up-to-date tally of the number of Flags a Post is receiving, and displaying an accurate count to Users. Ensure that the Post is automatically deleted after receiving 5 Flags.

2.13.5 Story Assigned to

- TBD

2.13.6 Story Points

- 2 Point (1 hour of work)

2.13.7 Status: Completed or not

- Not Completed. Plan of Action: to complete concurrently with other Posting Functionalities, specifically after both Create Post and Delete Post have been completed. Additionally, the number of Flags required to automatically delete a Post is subject to change and later will likely be tied to a percentage of the User-Base.

2.14 14 POST - Watchlist Post

2.14.1 Story Description:

- When a user wants to show support for a restaurant and give that restaurant preference in their notification settings.

2.14.2 Story Acceptance Criterion

- Watchlisting a post will work properly when the implementation successfully updates the post's user graphic interface and properly changes the user's notification preference settings.

2.14.3 Story Dependencies

- This story is dependent upon a valid user login, a verified education email, and a valid connection to firebase.

2.14.4 Story Challenges

- Firebase support and editing permissions for the notification preferences

2.14.5 Story Assigned to

- TBD

2.14.6 Story Points

- 2 Points (1 hour of work)

2.14.7 Status: Completed or not

- Not Completed. Plan of Action: to complete this story concurrently with the other POST stories

2.15 15 POST - Remove Watchlist Post

2.15.1 Story Description:

- When a user no longer wants to be notified about the post

2.15.2 Story Acceptance Criterion

- Removing a post from the user watchlist will work properly when the implementation successfully updates the post's user graphic interface and properly changes the user's notification preference settings.

2.15.3 Story Dependencies

- This story is dependent upon a valid user login, a verified education email, and a valid connection to firebase.

2.15.4 Story Challenges

- Firebase support and editing permissions for the notification preferences

2.15.5 Story Assigned to

- TBD

2.15.6 Story Points

- 2 Points (1 hour of work)

2.15.7 Status: Completed or not

- Not Completed. Plan of Action: to complete this story concurrently with the other POST stories

2.16 16 POST - Delete Post

2.16.1 Story Description:

- The delete a post functionality is for when a host wants to delete a post because of a canceled event or error in the posts general information

2.16.2 Story Acceptance Criterion

- Delete a post will work properly when the delete button is selected by the post owner and there is established connection with the Post Manager to verify the post owner's permissions and delete the post.

2.16.3 Story Dependencies

- This story is dependent upon a valid user login, a verified education email, and a valid connection to the Post Manager

2.16.4 Story Challenges

- Instantaneous access to the Post Manager

2.16.5 Story Assigned to

- TBD

2.16.6 Story Points

- 3 Points (1 and 1 half hours of work)

2.16.7 Status: Completed or not

- Not Completed. Plan of Action: to complete this story concurrently with the other POST stories

2.17 17 POST - Rate Post

2.17.1 Story Description:

- The rate a post functionality is for when a user wants to rate a post on how good ("hot") of a deal the post is

2.17.2 Story Acceptance Criterion

- Rate a Post will work properly when there is a 1-5 rating system (and set of buttons) implemented to allow the Users to choose if the Post was not a good deal (choose 1) up to the Post being a great deal (choose 5).

2.17.3 Story Dependencies

- This story is dependent upon a valid user login, a verified education email, and a valid connection to the Post Manager

2.17.4 Story Challenges

- Instantaneous access to the Post Manager

2.17.5 Story Assigned to

- TBD

2.17.6 Story Points

- 4 Points (2 hours of work)

2.17.7 Status: Completed or not

- Not Completed. Plan of Action: to complete this story concurrently with the other POST stories

2.18 18 BROWSE - Get Post List

2.18.1 Story Description:

- The client-side android application shall be able to query the firebase database for a list of posts, and is returned a list of posts with their respective data.

2.18.2 Story Acceptance Criterion

- A list of posts is returned to the querying browse function
- Each post contains pertinent data to be displayed as a scrollable list (Event name, rating).

2.18.3 Story Dependencies

- Firebase database
- POST-Creation

2.18.4 Story Challenges

- None

2.18.5 Story Assigned to

- Benjamin Miller

2.18.6 Story Points

- 8 points (4 hours of work)

2.18.7 Status: Completed or not

- Not completed. Plan of Action: to complete after VIEW - Browse is completed, also concurrently with other BROWSE stories

2.19 19 BROWSE - Sort Post List

2.19.1 Story Description:

- The post list used in BROWSE functionality may be sorted according to deal quality, proximity, or event date.

2.19.2 Story Acceptance Criterion

- Post list is sorted according to the description

2.19.3 Story Dependencies

- Firebase database backend support for Posts
- BROWSE-Get Post List

2.19.4 Story Challenges

- Efficiency is impacted by the availability of resources on the client application

2.19.5 Story Assigned to

- TBA

2.19.6 Story Points

- 2 points (1 hour of work)

2.19.7 Status: Completed or not

- Not completed. Plan of Action: to complete after VIEW - Browse is completed, also concurrently with other BROWSE stories

2.20 20 BROWSE - Filter Post List

2.20.1 Story Description:

- Post Manager is able to filter out posts from a list of posts that do not match the filter criteria

2.20.2 Story Acceptance Criterion

- Returned post list only contains posts that match the filter criteria

2.20.3 Story Dependencies

- Firebase database Post support
- BROWSE- Get Post List

2.20.4 Story Challenges

- Front end client resources

2.20.5 Story Assigned to

- TBA

2.20.6 Story Points

- 4 points (2 hours of work)

2.20.7 Status: Completed or not

- Not Completed. Plan of Action: to complete after VIEW - Browse Main and VIEW - Browse Filter are completed, also concurrently with other BROWSE stories

2.21 21 BROWSE - Back to Top Button

2.21.1 Story Description:

- A button on the Browse tab that returns the user to the beginning of the list

2.21.2 Story Acceptance Criterion

- A button will appear while scrolling on the browse tab that, when pressed the user's view will return to the top of the list

2.21.3 Story Dependencies

- BROWSE- Get Post List

2.21.4 Story Challenges

- Designing the user interface such that the button with be more of an aid than distraction

2.21.5 Story Assigned to

- TBD

2.21.6 Story Points

- 2 points (1 hour of work)

2.21.7 Status: Completed or not

- Not completed. Plan of Action: to complete after VIEW - Browse is completed, also concurrently with other BROWSE stories

2.22 22 BROWSE - Open Post

2.22.1 Story Description:

- While browsing, the user may press a button on each post labeled "more"
- This opens a temporary page that displays all details of event being posted about, such as Event Name, Event Host, Location/ Venue, Deal Rating, Number of Likes, Number of dislikes
- The page also contains a window that displays a google-maps image of the Event's location
- User may close the opened post and return to browse

2.22.2 Story Acceptance Criterion

- Function never causes mobile application to crash
- All information specified is listed on the page as specified
- This function is BROWSE or SEARCH agnostic, it will be used in either functionality

2.22.3 Story Dependencies

- BROWSE- Get Post List
- LOCATION SERVICES- Display Map
- POST- Create Post
- Firebase Database Post Support

2.22.4 Story Challenges

- This programming intensive as well as design intensive. Has many dependencies that need to couple properly to perform as expected.

2.22.5 Story Assigned to

- TBD

2.22.6 Story Points

- 8 Points (4 hours of work)

2.22.7 Status: Completed or not

- Not completed. Plan of Action: to implement concurrently with the views for post creation, posted event, and browse main are available paths into and out of this feature

2.23 23 LOCATION SERVICES - Directions to Restaurant/Event

2.23.1 Story Description:

- Using a get directions button on the post currently being navigated, directions are generated using the address provided in the post using Google's map services.

2.23.2 Story Acceptance Criterion

- Established connection to Google's map services and access to the address provided by the event host

2.23.3 Story Dependencies

- Post Manager access
- Embedded API provided by Google Maps

2.23.4 Story Challenges

- POST- Create Post
- VIEW- Posted Event

2.23.5 Story Assigned to

- TBD

2.23.6 Story Points

- 8 Point (4 hours of work)

2.23.7 Status: Completed or not

- Not Completed. Plan of Action: to implement after the button trigger is available in the post view

2.24 24 VIEW - Posted Event

2.24.1 Story Description:

- View for the mobile android application of the post details for the BROWSE- Open Post “Page” story

2.24.2 Story Acceptance Criterion

- View presents Post information in a clear and appealing way, with text details at the top with a map at the bottom

2.24.3 Story Dependencies

- POST- Create Post
- BROWSE- Get Post List
- BROWSE- Open Post

2.24.4 Story Challenges

- See challenges associated with BROWSE- Open Post

2.24.5 Story Assigned to

- TBD

2.24.6 Story Points

- 3 Points (1 and a half hours of work)

2.24.7 Status: Completed or not

- Not Completed. Plan of Action: Complete concurrently with BROWSE-Open Post.

2.25 25 VIEW - Post Creation

2.25.1 Story Description:

- This view focuses on the main attributes needed for post creation so that a host can post their event and the information affiliated with the event.

2.25.2 Story Acceptance Criterion

- A view of a large textbox for the main post and smaller labeled textboxes for important information such as the address and event title.

2.25.3 Story Dependencies

- ACCOUNT- Registration
- POST- Create Post

2.25.4 Story Challenges

- Adding a new post to the database using the Post Manager and Firebase

2.25.5 Story Assigned to

- TBD

2.25.6 Story Points

- 4 Points (2 hours of work)

2.25.7 Status: Completed or not

- Not Completed. Plan of Action: To implement prior to post creation to allow for editing of created posts by host users

2.26 26 VIEW - Browse Filter

2.26.1 Story Description:

- View for the mobile android application of the browse filter options

2.26.2 Story Acceptance Criterion

- Accepts the filters that the User chooses to apply to their Post List, allowing the User to customize their experience

2.26.3 Story Dependencies

- BROWSE - Filter Post LIst

2.26.4 Story Challenges

- Making an intuitive filter display for the User to understand and incorporate into their Cheap Eats experience.

2.26.5 Story Assigned to

- TBD

2.26.6 Story Points

- 6 Points (3 hours of work)

2.26.7 Status: Completed or not

- Not Completed. Plan of Action: to complete concurrently with the BROWSE - Filter Post List story.

2.27 27 VIEW - User Profile

2.27.1 Story Description:

- View of a user’s profile, displaying the user’s preferences, favorite hosts, favorite venues, and profile picture

2.27.2 Story Acceptance Criterion

- The view displays all data previously
- It does not crash the application

2.27.3 Story Dependencies

- ACCOUNT- User Registration

2.27.4 Story Challenges

- Retrieving data large amounts of specific data from the Firebase

2.27.5 Story Assigned to

- TBD

2.27.6 Story Points

- 2 Points (1 hour of work)

2.27.7 Status: Completed or not

- Not Completed. Plan of Action:

2.28 28 VIEW - Browse Main

2.28.1 Story Description:

- The view of all verified postings for events including the posts corresponding event title and restaurant name in clean graphics interface.

2.28.2 Story Acceptance Criterion

- The view displays all appropriate event titles and restaurant name

2.28.3 Story Dependencies

- BROWSE - Get Post List

2.28.4 Story Challenges

- Quick load time of the post in their preferred weighted order

2.28.5 Story Assigned to

- TBD

2.28.6 Story Points

- 3 Points (1 hour and 1 half hour of work)

2.28.7 Status: Completed or not

- Not Completed. Plan of Action: to complete concurrently with the BROWSE - Get Post List story.

2.29 29 VIEW - User Registration

2.29.1 Story Description:

- The View the user sees upon registering a new account

2.29.2 Story Acceptance Criterion

- The View displays all fields required to register a new user
- The View handles any verification properly
- Errors that occur during registration are displayed clearly without any external consequences

2.29.3 Story Dependencies

- ACCOUNT- Account Creation

2.29.4 Story Challenges

- None

2.29.5 Story Assigned to

- TBD

2.29.6 Story Points

- 2 Points (1 hour of work)

2.29.7 Status: Completed or not

- Not Completed. Plan of Action: Complete immediately by the end of the sprint 2 period, concurrently with Account Creation

2.30 30 VIEW - User Login

2.30.1 Story Description:

- Login view is strictly a view point for the user to see when they are attempting to login. This story is designed to help make the user's experience of getting into the application easier.

2.30.2 Story Acceptance Criterion

- The Login view is dependent upon reaching the firebase and sending and receiving request with that database. So the acceptance criteria for login will be dependent on the user having a valid connection to the database to process the login view pre generated from firebase.

2.30.3 Story Dependencies

- This story is dependent upon User Login; because this works hand in hand with each other. User Registration; because they must have a registered account to show the login view and same with Account Creation.

2.30.4 Story Challenges

- Some challenges faced with this view will be getting the login view to adapt to different types of phones / screens that will be accessing it. In other words, making it scalable and a variety of different hardware will be the main obstacle that we will have to overcome.

2.30.5 Story Assigned to

- TBD

2.30.6 Story Points

- 1 Point (1 half hour of work)

2.30.7 Status: Completed or not

- Not Completed. Plan of Action: to do implementation of this along with the other account functionality, specifically with ACCOUNT - User Login

2.31 31 SEARCH - Keyword Browse

2.31.1 Story Description:

- User may search posts by keyword
- A list of posts that match the provided keywords is returned

2.31.2 Story Acceptance Criterion

- A list of posts that match provided keywords is returned
- User may apply any BROWSE functionality to a searched list
- The list returned may have no posts

2.31.3 Story Dependencies

- BROWSE- Get Post List
- Firebase database post support
- Firebase keyword search

2.31.4 Story Challenges

- This leverages a new section of firebase’s functionality whose intensity in implementation is difficult to predict. Therefore it is safer to assume this will be intensive to implement

2.31.5 Story Assigned to

- TBD

2.31.6 Story Points

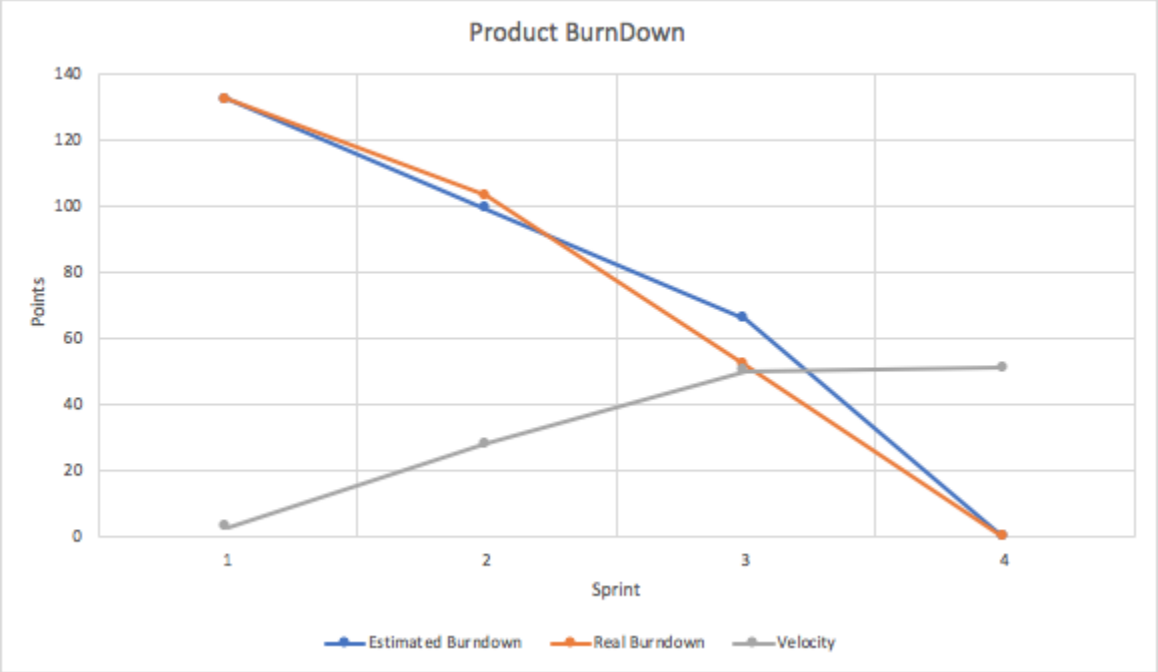
- 6 Points (3 hours of work)

2.31.7 Status: Completed or not

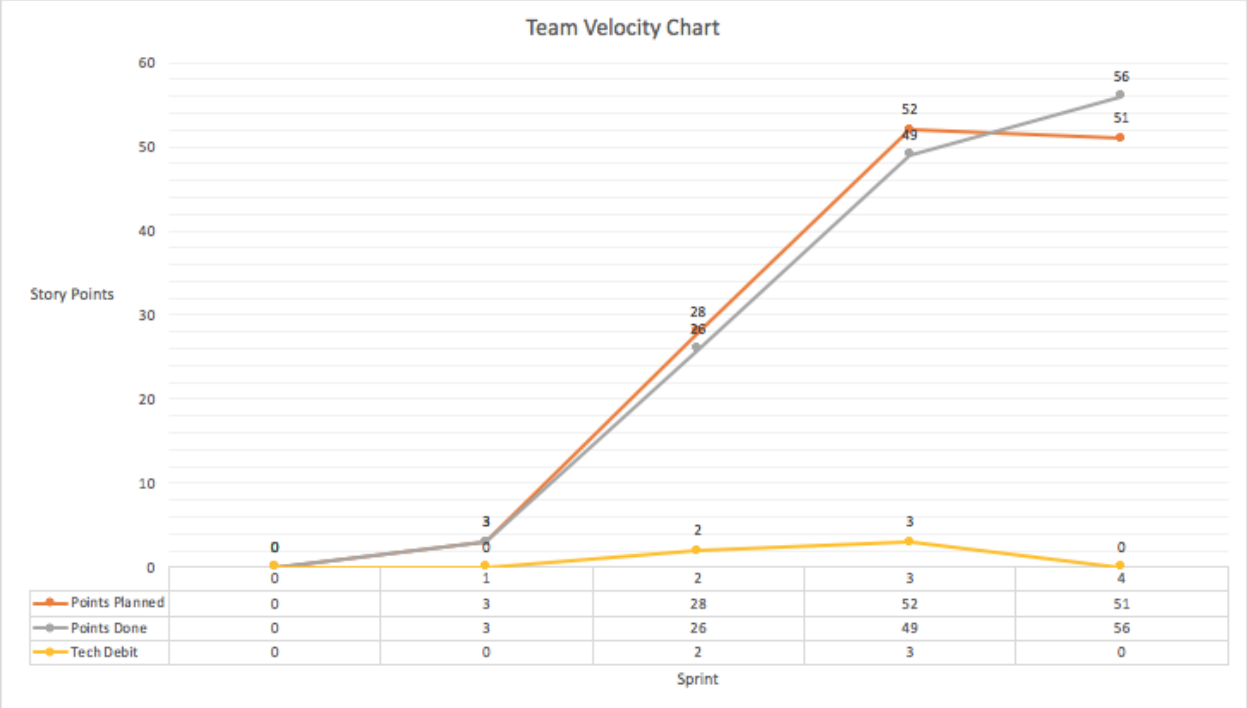
- Not Completed. Plan of Action: Implement after BROWSE functionality has been completed

3. Analytics

3.1 Sprint/Product Burndown Chart



3.2 Sprint Velocity



4. Conclusion

Sprint 1 has been a turning point for our group and the Cheap Eats Application. Scheduling conflicts have exacerbated the difficulties that arose when setting up the Android Studio SDK and environment, as well as general tooling difficulties. However, scheduling and general planning is going more smoothly, the further into the Sprints we are.

We foresee a quick adoption of both Android Studio and Google Firebase in the very-near future. With these adoptions, our progress will increase and the direction of a project will tend in a much better direction.