Software Engineering (CS-UH 2012)

Group Assignment 2

(Grocery on Rails)

Prepared by

Omar Kallas
Phillip Wang
Zayd Maradni
Zyad Yasser

Table of Contents

User Stories

User Story ID	Requirement	User Story
1	Sign-up for new customers	As an <online customer> who doesn’t have an account on the website, I want to be able to create an account by entering my email and choosing a password.
2	Sign-in for already existing customers	As an <online customer> who has an account on the website, I want to be able to log into my account easily by typing my email and password
3	Modify Account	As an <online customer> I want to be able to add and modify information on my account such as email, password, name
4	Multiple Addresses	An an <online customer> I want to be able to save multiple addresses on my account so I can have my groceries delivered to different places
5	Search box	As an <online customer> I want to be able to search for products using their name or a keyword related to them
6	Search filters	As an <online customer> I want to be able to filter my search results according to price, category, producing company, or other filters.
7	Categories view	As an <online customer> I want to be able to select from different categories of products.
8	Product view	As an <online customer> I want to be able to view the information about a product by clicking on it. Information includes price, company, number of items in stock, reviews, and estimated time of delivery.
9	Adding product to shopping cart	As an <online customer> I want to be able to add a product to my shopping cart so I can check out multiple items together. I want to add multiple items of the same product at once.
10	Viewing shopping cart	As an <online customer> I want to be able

		to view my shopping cart and the price of each product in it with the total price and an option to check out
11	Modifying shopping cart	As an <online customer> I want to be able to remove products from my shopping cart, or change the number of items of a certain product.
12	Viewing best deals	As an <online customer> I want to be able to view the new deals and offers on the website on the homepage to benefit from such deals
13	Viewing a customized list of items	As an <online customer> I want the homepage to include recommendations based on what I already bought to make my shopping experience faster.
14	Finalizing order	As an <online customer> I want to be able to finalize my order and enter a delivery address and estimated time, or have it scheduled to arrive at a different time, as I want my products fresh as possible.
15	Access order history	As an <online customer> I want to be able to view my previous purchases and the current status of all my orders
16	Payment	As an <online customer> I want to be able to make a secure payment through the website after I finalize my order. I want to be able to use widely used online payment such as Paypal, credit card, and local options. Also, I want to be able to pay with cash on delivery
17	Administrator login	As an <administrator> I want to be able to login through web browser into my account to be able to manage products
18	View and edit products database	As an <administrator> I want to be able to view the products database in a user-friendly GUI that allows me to see different information about products, modify them, remove products, or add new ones.
19	Out-of-stock notification	As an <administrator> I want to be able to see notifications through the web application when a product goes out of

		stock.
20	Create offers and discounted soon-to-expire products	As an <administrator> I want to be able to create special offers and deals and promote discounted soon-to-expire products to sell them quickly, and have that appear to the customers in the app home page.
21	View transactions	As an <administrator> I want to be able to view successful transactions with their status.
22	Contacting customers	As an <administrator> I want to be able to contact customers either directly through email provided or via newsletters
23	Backup and restore data	As an <administrator> I need to be able to backup and restore the database regularly and easily, as food products have expiry periods, so there are frequent changes to the database that need to be available
24	Print Invoices and packing lists	As an <administrator> I want to be able to print invoices and packing lists from the administrator anel both for individual orders and all current orders
25	Access to sale statistics	As an <administrator> I want to be able to access statistics for sales and what products are being bought especially for controlling stock of perishable items as relevant, and statistics for customers and their orders and purchases
26	Subscribing to a alerts and newsletters	As an <online customer> I want to be able to receive updates about certain products and their availability, receive notifications of deals or on discounted soon-to-expire products.
27	Availability of out-of stock items	As an <administrator> I want to be able to change whether an item that is out-of-stock or is expired can still be shown to the customer
28	SSL Transactions	As an <online customer> I want my transactions to use SSL to ensure the security of my private and payment information

29	Frequently Bought Together	As an <online customer> I want to see what other customers bought with this type of food to discover more options of food that I can prepare/consume together or learn from others what tastes good together.
30	Contacting the administrator	As an <online customer> I want to ask about more detailed information about the merchandise, talk to them about user experience and suggestions, and raise issues when the delivery goes wrong.
31	Return the order	As an <online customer> I want to return my orders and get a refund if the products have expired or they do not fit their description online.
32	View home screen	As an <online customer> I want to be able to view the home screen in the app, that includes new products, recommended products, and discounts

Table 1. Shows user stories of the backlog and their description

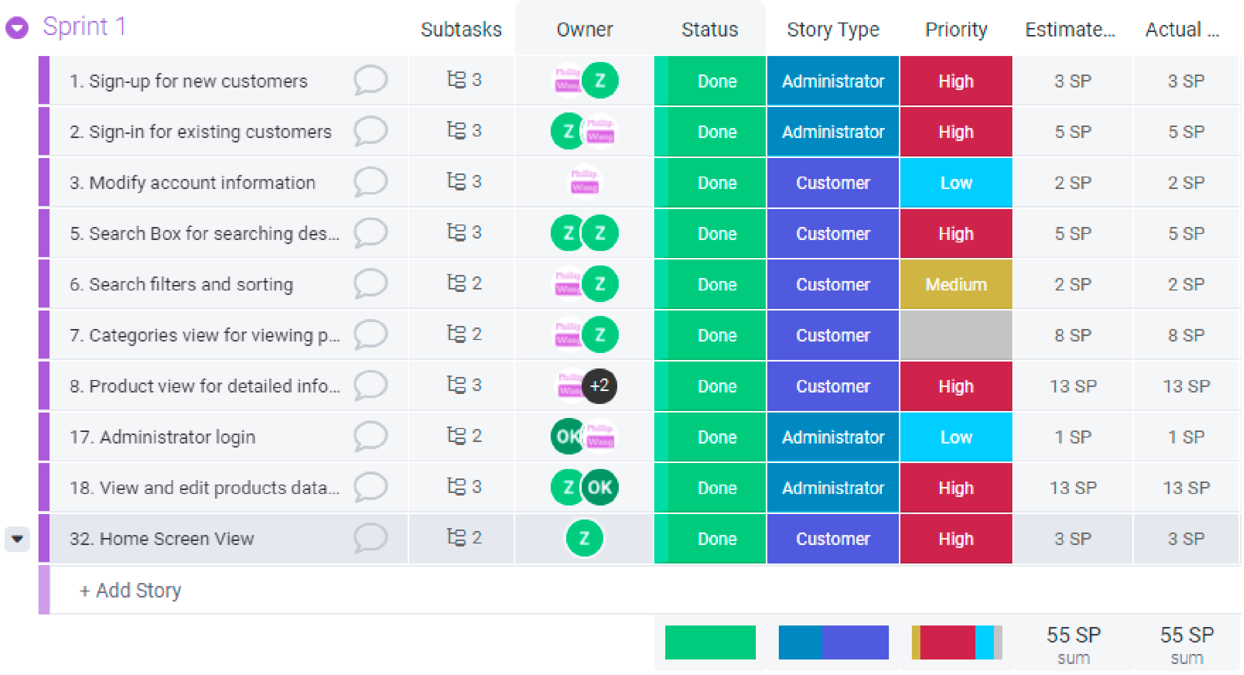
Project Backlog

Sprint 1 backlog was decided based on the priority and size of the tasks. We decided to implement the high priority tasks first, naturally, and the larger sized tasks, as those usually correspond to more important features of the system. We also focused on finishing the stories that are related to the infrastructure of the system, that stories that can be reused for later, and those that other stories depend on or that are small additions and feature enhancements on them. The number of stories was chosen with our assumption that 0.7 story point can be done per day per person, which roughly put our team capacity at 56 points, this will be modified as we gain more understanding of our tasks, how efficient we can be, what our actual velocity is and the nature of tasks, as larger and infrastructure stories might take longer and are more difficult than small feature additions. The tasks we have decided upon amount to 55 story points for sprint 1.

After spending lots of time learning and building the main infrastructure in Sprint 1, we realized that we could increase our velocity in sprint 2 from 0.7 to 0.85 story points per person per day. Another reason for that increase is that we underestimated our capacity before and realized that we could be more efficient. So we realized that our capacity can be increased from 55 to 65 story points. Most of these story points are divided among medium and low priority tasks.

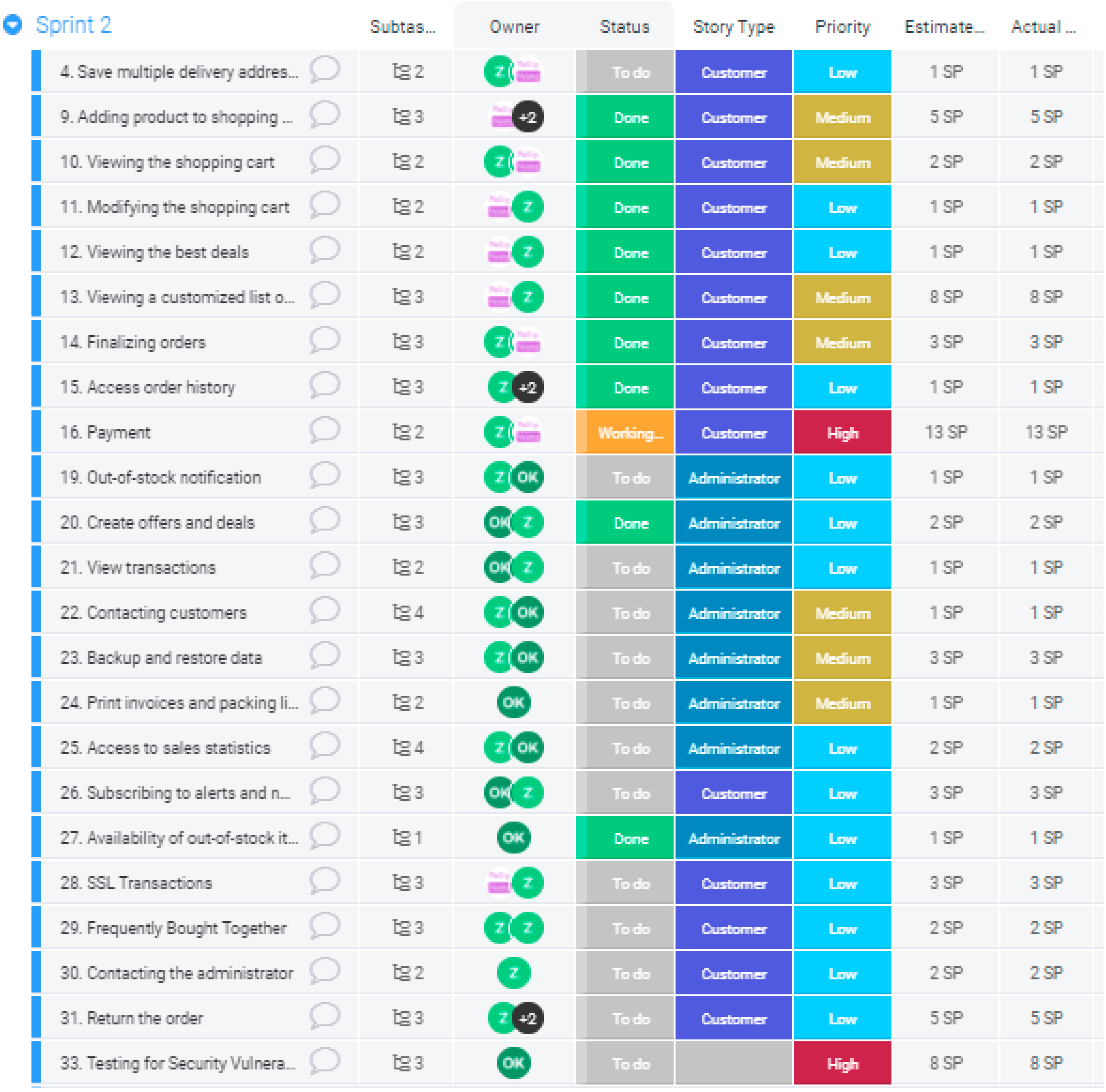
Product Backlog (from monday.com)

We used (monday.com) as our scrum planning platform. Below is a screenshot of the website that shows the project backlog/sprint planning.



Sprint 1	Subtasks	Owner	Status	Story Type	Priority	Estimate...	Actual ...
1. Sign-up for new customers	3	Phillip Wang, Z	Done	Administrator	High	3 SP	3 SP
2. Sign-in for existing customers	3	Z, Phillip Wang	Done	Administrator	High	5 SP	5 SP
3. Modify account information	3	Phillip Wang	Done	Customer	Low	2 SP	2 SP
5. Search Box for searching des...	3	Z, Z	Done	Customer	High	5 SP	5 SP
6. Search filters and sorting	2	Phillip Wang, Z	Done	Customer	Medium	2 SP	2 SP
7. Categories view for viewing p...	2	Phillip Wang, Z	Done	Customer		8 SP	8 SP
8. Product view for detailed info...	3	Phillip Wang, +2	Done	Customer	High	13 SP	13 SP
17. Administrator login	2	OK, Phillip Wang	Done	Administrator	Low	1 SP	1 SP
18. View and edit products data...	3	Z, OK	Done	Administrator	High	13 SP	13 SP
32. Home Screen View	2	Z	Done	Customer	High	3 SP	3 SP
+ Add Story							
						55 SP sum	55 SP sum

Figure 1. A screenshot of monday.com that shows the backlog of sprint 1



Sprint 2

	Subtas...	Owner	Status	Story Type	Priority	Estimate...	Actual ...
4. Save multiple delivery addres...	2	Z	To do	Customer	Low	1 SP	1 SP
9. Adding product to shopping ...	3	+2	Done	Customer	Medium	5 SP	5 SP
10. Viewing the shopping cart	2	Z	Done	Customer	Medium	2 SP	2 SP
11. Modifying the shopping cart	2	Z	Done	Customer	Low	1 SP	1 SP
12. Viewing the best deals	2	Z	Done	Customer	Low	1 SP	1 SP
13. Viewing a customized list o...	3	Z	Done	Customer	Medium	8 SP	8 SP
14. Finalizing orders	3	Z	Done	Customer	Medium	3 SP	3 SP
15. Access order history	3	Z +2	Done	Customer	Low	1 SP	1 SP
16. Payment	2	Z	Working...	Customer	High	13 SP	13 SP
19. Out-of-stock notification	3	Z OK	To do	Administrator	Low	1 SP	1 SP
20. Create offers and deals	3	OK Z	Done	Administrator	Low	2 SP	2 SP
21. View transactions	2	OK Z	To do	Administrator	Low	1 SP	1 SP
22. Contacting customers	4	Z OK	To do	Administrator	Medium	1 SP	1 SP
23. Backup and restore data	3	Z OK	To do	Administrator	Medium	3 SP	3 SP
24. Print invoices and packing li...	2	OK	To do	Administrator	Medium	1 SP	1 SP
25. Access to sales statistics	4	Z OK	To do	Administrator	Low	2 SP	2 SP
26. Subscribing to alerts and n...	3	OK Z	To do	Customer	Low	3 SP	3 SP
27. Availability of out-of-stock it...	1	OK	Done	Administrator	Low	1 SP	1 SP
28. SSL Transactions	3	Z	To do	Customer	Low	3 SP	3 SP
29. Frequently Bought Together	3	Z Z	To do	Customer	Low	2 SP	2 SP
30. Contacting the administrator	2	Z	To do	Customer	Low	2 SP	2 SP
31. Return the order	3	Z +2	To do	Customer	Low	5 SP	5 SP
33. Testing for Security Vulnera...	3	OK	To do		High	8 SP	8 SP

Figure 2. A screenshot of monday.com that shows the backlog of sprint 2

Poker Estimation Description

We followed a fibonacci style for deciding the story points, the sizes are 1,2,3,5,8,13. Each person estimates the size and reveals it to the rest. If they are not the same size, a discussion is held if someone believes their size is more accurate, these opinions can be from any person and not necessarily the person who will work on the story. This ensures that various perspectives are considered, especially if the person working on it hasn't considered them. But also, a person might have more expertise and if they can counter those arguments then their estimate is followed

User Story	Story Points	Rounds	Estimations in each round	Description
2	5	2	Omar: 5-5 Phillip: 5-5 Zayd: 2-5 Zyad: 3-5	Zayd thinks that a sign in feature shouldn't need much work other than direct database query for the user account. Phillip interjected with how it needs more time not just for developing the UI, but handling authentication and secure transmission, sign-in also includes returning tokens and saving informations into sessions to be facilitate ease of use, so it is a much larger task

Table 2 - Shows an example of the poker estimation

Sample table shows the user story id, the story points decided upon, estimation in each round and a description of how it was reached. For example, this is the estimation for the user story #2. The group decided to give it a size 5 after 2 rounds. The first round, Omar estimated 5, Phillip 5, Zayd 2, Zyad 3, while the second everyone estimated 5. The description includes the discussions that happened each round and why everyone gave the task that size. When there is a consensus in the first round there is a brief description of why it was reached. Otherwise, the main arguments and counter arguments are stated of why it should be a specific size. Some are decided upon after a general agreement, and some because the person working on it knows more about it or has more experience with it, and thus their expertise was agreed upon and their estimations held more weight.

User Story	Story Points	Rounds	Estimations in each round	Description
1	3	1	Omar: 3 Phillip: 3 Zayd: 3 Zyad: 3	Group agreed from round 1, other than the UI and input sanitation, the other queries and tasks are simple.

2	5	2	Omar: 5-5 Phillip: 5-5 Zayd: 2-5 Zyad: 3-5	Zayd thinks that a sign in feature shouldn't need much work other than direct database query for the user account. Phillip interjected with how it needs more time not just for developing the UI, but handling authentication and secure transmission, sign-in also includes returning tokens and saving informations into sessions to be facilitate ease of use, so it is a much larger task
3	2	1	Omar: 2 Phillip: 2 Zayd: 2 Zyad: 2	Relatively simple, needs only the frontend. The backend, sign in, input sanitation reused from sign up process and simple query to database
4	1	1	Omar: 1 Phillip: 1 Zayd: 1 Zyad: 1	Small task, Same as first address input and small query to database
5	5	2	Omar: 3-5 Phillip: 5-5 Zayd: 5-5 Zyad: 8-5	Zyad thinks that a search feature will take a long time, because it has to be established on the backend and frontend, and would require searching and showing all matching products. Phillip stated that the backend and requests are not that complicated to handle. While Zayd said that the actual search functionality is very easily handled by the database and the rest of the front-end is already implemented, so it only requires the UI for the search box
6	2	2	Omar: 1-2 Phillip: 3-2 Zayd: 2-2 Zyad: 3-2	Phillip wanted it to give it more time, since developing the front end for the filters might take some time, and there is much in the backend to be done to handle the logic of the filtering. Zayd explained how all the filtering can be handled through the database query, returning a filtered list of items, so the main work to be done is on the front end.
7	8	1	Omar: 8 Phillip: 8 Zayd: 8 Zyad: 8	There was consensus on this from the first round, since this mostly in the front end and it is a major and backbone functionality that other functionalities also depend on
8	13	2	Omar: 5-13 Phillip: 5-13 Zayd: 13-13	Phillip argued that the product page only need to display the information given by the backend which is a small task. Zyad points

			Zyad: 8-13	out that this an integral part of the app and need to be done properly, in addition to the information given by the backend is of various sizes and numbers and there are different tables to handle. The front end needs to be coded to be dynamic to handle all of these cases where there are multiple images, multiple description such as nutritional information, without giving errors. Zayd added that to have a product view, we need to have a database of the products, and to have that we need to identify websites to scrap from and do a proper scraping to get all the information including all the categories and subcategories of the products. This makes this the largest user story and possibly the most important one as it lays the basis for most of the other stories. The group agreed on the size 13 as one of the most important and largest stories
9	5	3	Omar: 2-3-5 Phillip: 5-5-5 Zayd: 3-5-5 Zyad: 3-3-5	Zyad explained how from the front end it doesn’t require much work after all the previous stories. Omar believed, while integral, a cart implementation is not very difficult from the backend, as it only needs to save the products and their info. Phillip disagreed, as a cart needs to be saved to the user’s session, and needs to be retrievable between sessions. After the second round, Phillip emphasized the logic that needs to go behind a cart, such multiple tests to ensure the availability of the products, putting a temporary hold on the amount of the cart so that other people can’t finish the stock before the user has had a chance to checkout, convincing the rest of the size 5
10	2	1	Omar: 2 Phillip: 2 Zayd: 2 Zyad: 2	Consensus, since it mostly requires the development of the front end for the cart.
11	1	1	Omar: 1 Phillip: 1 Zayd: 1 Zyad: 1	Consensus, since most of the tests should be implemented in a previous story, so it only requires a small addition to it plus the UI.
12	1	2	Omar: 3-1 Phillip: 2-1	The three members decided that it would take a while to be able to add the frontend to

			Zayd: 2-1 Zyad: 1-1	handle tabation, but Zyad said that he has experience in it and that it is mostly a reuse of the products view with an added tabation
13	8	3	Omar: 2-5-8 Phillip: 5-8-8 Zayd: 13-13-8 Zyad: 3-5-8	Zayd explained that the task is larger than it looks. Recommending products to users requires going through the user’s previous searches, viewing what other people have purchased, matching products that are mostly bought together, identifying what makes products suitable for the customer it is not a simple returning products from a similar category or name, and this would require some proper data science, giving it a 13. The group then settled on 8 after deciding that it is a large task, but not very important to go into as much depth as Zayd proposed, but not as shallow as they first thought, matching similarity in category, name,.brand. Further discussion of why a proper recommendation system is still important to do led to the estimation of 8
14	3	2	Omar:2-3 Phillip: 3-3 Zayd: 2-3 Zyad: 5-3	Zyad proposed that the UI for this will need some time to do, other than the backend. Zayd argued that it is only a form field and shouldn’t need time to just input values and send them to the backend. Phillip then talked about how while the UI might be simple, the backend needs more work to do input sanitation, check for validity of address, delivery range, accessing old addresses, The group agreed on size 3
15	1	1	Omar: 1 Phillip: 1 Zayd: 1 Zyad: 1	Already saved and needs a simple query to return the results
16	8	3	Omar: 13-13-8 Phillip: 5-8-8 Zayd: 13-8-8 Zyad: 13-13-8	Most of the group saw that this will be a very large task, given the different payment methods credit cards, paypal, vouchers,... While Phillip said that he had experience with this and that there are already-made libraries for them so it only needs a 5. The group was still convinced that this feature would still require a lot of testing and there are major areas for bugs and issues, and also it will be time consuming to test for all the different platforms and work the quirks of each one.

				Eventually, the group decided on size 8, as a large task but the libraries and Phillip's experience should cut part of the size.
17	1	1	Omar:1 Phillip: 1 Zayd: 1 Zyad: 1	Consensus, given that admin logging will be done after the user login, so the framework only needs slight changes to work for the admin.
18	13	3	Omar: 13-13-13 Phillip: 5-8-13 Zayd: 8-8-13 Zyad: 8-13-13	Phillip and Zayd both thought that this wouldn't require much time, given that the backend basis was already established for the customer. Omar and Zyad disagreed, as the admin portal still needs to be fully designed, and the website format means there will be not much reuse from the mobile app. Further discussions in the rounds showed that the backend still needs a lot of work given that it will enable changes to the database that would require their own logic and input checking. In addition to the front end in this user story being the main backbone of the website that will be reused for the other functionalities, so work must be put in it to be reusable and robust. The team decided on size 13 for the amount of work and design needs to be put in
19	1	1	Omar:1 Phillip: 1 Zayd: 1 Zyad: 1	Everyone agreed that this is a small task that only needs to check if an item is out of stock after every purchase
20	2	2	Omar: 2-2 Phillip: 3-2 Zayd: 1-2 Zyad: 5-2	Zyad pointed out that this would need a special frontend page and behavior to account for the offer. Omar argues that the frontend is mostly implemented, with small work in the backend, since it only needs a new price and to be put as a promotion or discounted soon-to-expire
21	1	1	Omar: 1 Phillip: 1 Zayd: 1 Zyad: 1	Small task, needs to be retrieved from the database
22	1	1	Omar: 1 Phillip: 1 Zayd: 1 Zyad: 1	Easy task, only need to access user's email from the database, and query users that are subscribed to the newsletter

23	3	2	Omar: 1-3 Phillip: 2-3 Zayd: 5-3 Zyad: 3-3	Zayd believed that this a relatively difficult task to find a good frequency for the backups, ensuring they are working. Phillip pointed out that a lot of the work can be handled automatically by the hosting site. Zayd also noted that this still needs a lot of testing to ensure that it works, and is practical space/time wise. They reached a compromise of 3
24	1	1	Omar: 1 Phillip: 1 Zayd: 1 Zyad: 1	Small task, the information already provided, only a needs a small function to print the page’s data
25	2	2	Omar: 5-2 Phillip: 5-2 Zayd: 2-2 Zyad: 3-2	Omar discussed how building such statistics will be complicated, having to scan through the database and aggregating multiple information in multiple ways depending on what the admin wants to display. Zayd pointed out that it is actually not that difficult, since the statistics will be automatically updated after every purchase, and all the smaller details will be saved in the order history and products, accessing the different statistics only needs slightly different aggregate queries that are easy to write.
26	3	2	Omar: 2-3 Phillip: 2-3 Zayd: 3-3 Zyad: 3-3	Phillip states that this is rather easy, as we just need to query the users subscribed and then send the newsletter when it is written. Zayd points out that it is more complicated, since users can also place alerts for certain products and the newsletter needs to be curated ot the user, so it needs more time to work out those details
27	1	1	Omar: 1 Phillip: 1 Zayd: 1 Zyad: 1	Simple check box to control visibility
28	3	2	Omar: 2-3 Phillip: 1-3 Zayd: 2-3 Zyad: 3-3	Phillip said that there are libraries and standards for SSL, so it doesn’t require much work to get it up. Zayd mentions that it would still need more testing to ensure that it is correct and no issues.
29	2	1	Omar: 2 Phillip: 2	Zayd said that it is relatively small, since we have access to all orders and we simply

			Zayd: 2 Zyad: 2	aggregate all the orders with that product and sort them.
30	2	2	Omar: 2-2 Phillip: 4-2 Zayd: 3-2 Zyad: 2-2	Phillip and Zayd both think that even though the functionality is not complicated, we need to manage the chat history and be able to link it to a specific entry in the database. Omar suggests that we could use third-party tools to integrate into the system that handles virtually everything for us
31	5	1	Omar: 5 Phillip: 5 Zayd: 5 Zyad: 5	Everyone agrees that the task is relatively harder since it involves in changing the state of an order and we need to follow-up on both the customer and the administrator on returning the order
32	3	1	Omar: 3 Phillip: 3 Zayd: 3 Zyad: 3	Everyone agrees that the task is simple since it’s really similar to the category view page so it will require just some slight modifications.
33	8	2	Omar: 13-8 Phillip: 5-8 Zayd: 8-8 Zyad: 13-8	Everyone agrees that the task is complicated since security vulnerabilities are hard to check and find despite being very critical to eliminate.

Table 3 - shows the poker estimation and the discussion for all the user stories

Team Operation Protocols

Procedure for Daily Scrum Meeting

The daily scrum meeting is limited to 10 minutes so that the team members will not be trapped in meetings. The Scrum Master should time the session and lead the discussion. The first round of discussion begins with everyone briefing their progress yesterday. Should any member experience obstacles during the development, concerns should be raised and briefly discussed by the team. The second round of discussion starts with team members claiming their tasks to focus on today and requesting necessary resources or personnel from the team to complete the tasks. The SCRUM master should make sure that the discussion does not deviate from the summary and planning of tasks and should arrange additional meetings for issues that require more time. Additionally, if any user stories are completed, the owner of the story should demo the functionality during the daily scrum meeting and continue working on it if other members and Product Owner have any reasonable suggestions.

Protocols for Resolving Disagreements

When a disagreement arises regarding the implementation details of a task, both sides should present their arguments in the communication channel, enunciating their rationale and reasoning behind their arguments. The other members are welcome to add their comments but all discussion should be backed with evidence and refrain from insults and sarcasm. The team will eventually vote on the issue if both sides insist on their opinion but the product owner has the right to overrule the decision. However, both opinions should be archived for future reference and they can be re-examined to make sure that the team made the correct decision or the team learns from a wrong decision.

Work Guidelines

- Scrum Master should aware of any vacation days as soon as possible, no later than 48 hours before
- As the team members are in difference time zones, do not expect immediate response from teammate in the communication tool but always leave your questions in the channel so that everybody can see the question and the answer to it
- When the code is not working as expected, try checking and testing your own code first, then troubleshooting the problem with the team and invite others to test the code in different devices
- Always look for implemented libraries to use instead of writing code from scratch.

Sprint 1

Sprint goals

Build the main skeleton of the shopping system, by finishing the products and categories database, and connecting it to the mobile application allowing the user to have search functionality and the ability to view products and their information. The backend built in this sprint should be flexible and dynamic to allow us to add the rest of the user's features and the administrator's features in the next sprint easily without much modification.

Team working agreement

Ceremonies	
Stand Up	Daily start at 10am
Working Hours	6 hours a day, from Sunday until Thursday
Iteration scrum meeting	10 mins daily at 10am
Roadmap planning	Twice a week
Duration	Mar. 7 - Apr. 1
Scrum Master	Omar
Product Owner	Zyad
Communication tools	
Project Discussion	Messenger/Slack
Document sharing	Google docs
Code sharing	Github (https://github.com/grocery-on-rails/)
Object	
Sprint 1 object	The user's mobile app should be ready with search and product viewing features. The database and backend should be in place and ready to be modified for the next sprint
Test object	Search function returns relevant results, no bugs in viewing products
Demo and Review	Demo to product owner

Communication guidelines	
Communication before starting a task	Each team member should inform the rest of the team before starting a task, to make sure that the backend and frontend are on the same page, and to be informed in case there are specific guidelines/specifications that were assumed by other members when completing a task connected to this one.
Communication during a task	A team member shouldn’t hesitate to ask others for clarification or help if they think something is unclear to them. Clarifying at an early stage is better than making assumptions that could result in bugs and miscommunication later.
Communication after finishing a task	After a member finishes a task, they should inform the rest of the members to check the compatibility of the backend and the frontend of that task, and to allow other members to start working on tasks dependent on that task.

Table 4 - shows the team working agreement and its details

The Definition of Done

- Finish the database of products and categories.
- Finish the mobile application with the home screen, product view, and search functionality.
- A feature is done only when both backend and frontend are connected and operating together.
- Test the different functionalities implemented in this sprint.
- Demo the function of the online shopping system
  .

Assumptions

1. 2.75 story point per day on average
2. If a user story had multiple tasks assigned to different team members, the story points were split accordingly.
3. In this sprint, developers will need time to familiarize themselves with the technologies and tools used.
4. Developers will be available during their working hours to answer or clarify any questions.
5. Team members will be working on the project until the completion of the system (none of them will have an unexpected circumstance that would force them to leave).

Tasks for Each User Story in Sprint 1 Backlog

Title	Task 1	Task 2	Task 3
1. Sign-up for new customers	Write the needed query and needed checks for adding a new user	Implement /auth/signup api that registers a new user with provided credentials	Check whether provided email is already registered in database
2. Sign-in for existing customers	Implement /auth/signin that returns a token for identification	Check password against the hashed string in database	Check whether the email is registered
3. Modify account information	Update account password or email	Check user identity before updating the information	Designing the UI
5. Search Box for searching desired products with keywords	Create Search Index	Implement search logic in backend	Designing the UI
6. Search filters and sorting	Edit the query to allow for sorting and filtering	Unit test	
7. Categories view for viewing products with the same category	Implementing GET /cat/ for getting the list of categories	Designing the categories view page	
8. Product view for detailed info about a product	Scrap website to get database data	Design the database and populate the data	Implement the needed queries in the backend
17. Administrator login	Implement /auth/signin for admin users	Design the website for admin to login	
18. View and edit products database	Implement logic and queries in the backend	Write needed checks for the input before modification	Unit test, test input sanitation
32. Home Screen View	Designing the home screen UI	Integrating /GET/ home endpoint to get the data	

Table 5 - shows the user stories in Sprint 1 Backlog, and the tasks each story is divided into

Daily Sprint 1 Report

The daily sprint report details each day, what stories did the members work on and how many story points they finished on that day. In addition, it also shows their productivity on that day, where 100% means finishing 1 point on a given day. The notes show and overview of what happened on that day.The table is color coded as follows:

Shows that the person was over-productive, achieving 100% or higher productivity on that day or the team burnt 3.25 or more points, someone finished a task (in the notes)

Shows that the person was under-productive, achieving 50% or less productivity on that day or that the team burnt 2.25 or less points

Shows that the person is done with the points for that story, or that a person is done with the sprint (in the notes)

Shows that the person took a vacation on that day

Date	Omar			Phillip						Zayd				Zyad						Productivity %					Notes
Story ID	18	17	32	8	5	2	1	7	3	8	5	1	18	8	5	7	6	3	32	Omar	Phillip	Zayd	Zyad	Burnt Points	
Story Points	10	1	2	2	1	5	2	3	1	5	1	1	3	6	3	5	2	1	1						
March 7 2021	0.5	0	0	0.1	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	50	10	100	40	2.0	
March 8 2021	1.4	0	0	1.3	0.0	0.0	0.0	0.0	0.0	1.8	0.0	0.0	0.0	1.8	0.0	0.0	0.0	0.0	0.0	90	120	80	140	4.3	
March 9 2021	1.9	0	0	2.0	0.1	0.0	0.0	0.0	0.0	1.9	0.0	0.0	0.0	2.7	0.0	0.0	0.0	0.0	0.0	50	80	10	90	2.3	Phillip done 8 started #5
March 10 2021	2.5	0	0	2.0	1.0	0.0	0.0	0.0	0.0	2.2	0.0	0.0	0.0	2.8	0.0	0.0	0.0	0.0	0.0	60	90	30	10	1.9	Phillip finished #5
March 11 2021	**2.5**	0	0	2.0	1.0	1.5	0.0	0.0	0.0	2.8	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0	150	60	120	3.3	Phillip started #2 Omar took a vacation
March 14 2021	2.8	0	0	2.0	1.0	1.8	0.0	0.0	0.0	3.0	0.0	0.0	0.0	5.4	0.0	0.0	0.0	0.0	0.0	30	30	20	140	2.2	
March 15 2021	4.0	0	0	2.0	1.0	3.0	0.0	0.0	0.0	3.6	0.0	0.0	0.0	6.0	0.5	0.0	0.0	0.0	0.0	120	120	60	110	4.1	Zyad finished 8 and started #5
March 16 2021	5.1	0	0	2.0	1.0	4.1	0.0	0.0	0.0	3.8	0.0	0.0	0.0	6.0	1.0	0.0	0.0	0.0	0.0	110	110	20	50	2.9	
March 17 2021	6.2	0	0	2.0	1.0	4.4	0.0	0.0	0.0	4.3	0.0	0.0	0.0	6.0	2.3	0.0	0.0	0.0	0.0	110	30	50	130	3.2	

March 18 2021	7.2	0	0	2.0	1.0	**4.5**	0.0	0.0	0.0	5.0	0.8	0.0	0.0	6.0	2.6	0.0	0.0	0.0	0.0	100	10	150	30	2.9	Zayd finished #8 and started #5
March 21 2021	7.8	0	0	2.0	1.0	4.7	0.0	0.0	0.0	5.0	1.0	0.0	0.0	6.0	3.0	0.8	0.0	0.0	0.0	60	20	20	120	2.2	Zayd finished #7 and started #5 Zyad finished #5
March 22 2021	9.3	0	0	2.0	1.0	5.0	1.0	0.0	0.0	5.0	1.0	0.7	0.0	6.0	3.0	1.5	0.0	0.0	0.0	150	130	70	70	4.2	Phillip finished #2 and started #1
March 23 2021	10.0	0	0	2.0	1.0	5.0	1.4	0.0	0.0	5.0	1.0	1.0	1.1	6.0	3.0	2.3	0.0	0.0	0.0	70	40	140	80	3.3	Omar finished #18 Zayd finished #1 started 18
March 24 2021	10.0	1	0.3	2.0	1.0	5.0	1.7	0.0	0.0	5.0	1.0	1.0	2.1	6.0	3.0	2.8	0.0	0.0	0.0	130	30	100	50	3.1	Omar started #32
March 25 2021	10.0	1	1.2	2.0	1.0	5.0	2.0	1.2	0.0	5.0	1.0	1.0	3.0	6.0	3.0	4.2	0.0	0.0	0.0	90	150	90	140	4.7	Zyad finished #5 and started #6 Phillip finished #1 and started #7 Zayd done with Sprint 1
March 28 2021	10.0	1	1.9	2.0	1.0	5.0	2.0	2.7	0.0	5.0	1.0	1.0	3.0	6.0	3.0	5.0	0.2	0.0	0.0	70	150		100	3.2	
March 29 2021	10.0	1.0	2	2.0	1.0	5.0	2.0	3.0	0.7	5.0	1.0	1.0	3.0	6.0	3.0	5.0	1.2	0.0	0.0		100		100	2.1	Omar done with Sprint 1
March 30 2021	10.0	1.0	2	2.0	1.0	5.0	2.0	3.0	1.0	5.0	1.0	1.0	3.0	6.0	3.0	5.0	2.0	0.0	0.0		30		80	1.1	Phillip done With Sprint 1
March 31 2021	10.0	1.0	2	2.0	1.0	5.0	2.0	3.0	1.0	5.0	1.0	1.0	3.0	6.0	3.0	5.0	2.0	0.9	0.0				90	0.9	
April 1 2021	10.0	1.0	2	2.0	1.0	5.0	2.0	3.0	1.0	5.0	1.0	1.0	3.0	6.0	3.0	5.0	2.0	1.0	1.0				110	1.1	Everyone is done with Sprint 1. Completed all user stories

Table 6 - shows detailed report of sprint 1, including story points each person did,total points burnt on that day, member's productivity and overview of the events of each day

Burndown Chart

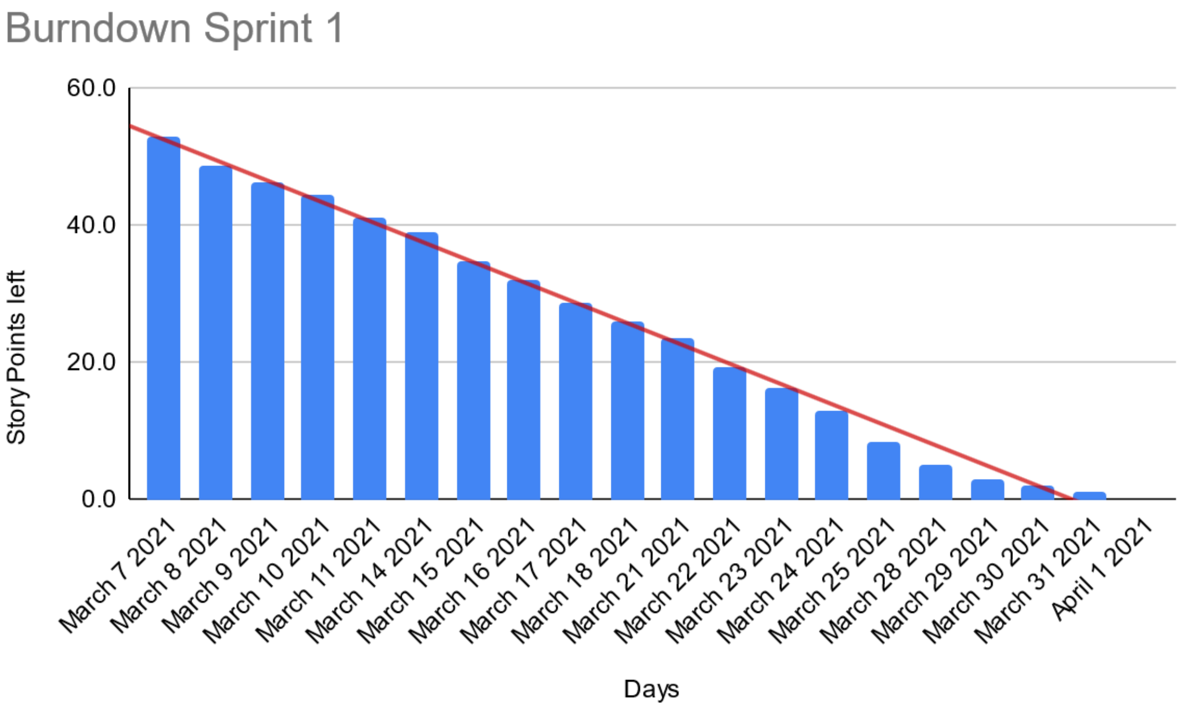


Chart 1. Shows the number of stories left per day during Sprint 1

Sprint Review

The team managed to finish all the stories for Sprint 1. The velocity was 55 points/sprint or 2.75 points per day.

Sprint Retrospective

The team generally did a good job during Sprint 1. Below is the summary of lessons learned discussed during the sprint retrospective meeting.

What went well?

- Each person learned the technologies they needed, so we didn’t have to learn everything.
- Division of tasks.
- Communication between frontend and back end

What went wrong?

- Learning new technologies took some extra time.
- Although communication before the frontend and backend was smooth, there wasn’t much planning before completing the tasks.
- No proper testing, only proof of concept.

Sprint 2

Sprint goals

Continuing to implement the functionalities and finalizing the whole project by building the constituent parts (backend, app, administrator panel). The main features that should be added are but not restricted to implementing the shopping cart feature, payment, order, order history, secure transaction. The administrator panel should be up and fully running by the end of the sprint to allow the administrator to view transactions, contact customers, print invoices, access statistics, and more. Also, during this sprint we should have a stressed emphasis on testing to produce a bug-free product.

Team working agreement

Ceremonies	
Stand Up	Daily start at 10am
Working Hours	6 hours a day, from Sunday until Thursday
Iteration scrum meeting	10 mins daily at 10am
Roadmap planning	Twice a week
Duration	April 2 - April 29
Scrum Master(s)	Phillip
Product Owner	Zayd
Communication tools	
Project Discussion:	Slack
Document sharing:	Google docs
Code sharing:	Github (https://github.com/grocery-on-rails/)
Object	
Sprint 1 object	The mobile app, administrator panel, DB, backend should be fully functional.
Test object:	● Every module should be tested individually. ● Integration testing should be used to eliminate the bugs resulting from integrating backend/app, backend/DB, backend/admin

Demo and Review	Demo to product owner
Communication guidelines	
Communication before starting a task:	Each team member should inform the rest of the team before starting a task, to make sure that the backend and frontend are on the same page, and to be informed in case there are specific guidelines/specifications that were assumed by other members when completing a task connected to this one.
Communication during a task:	A team member shouldn't hesitate to ask others for clarification or help if they think something is unclear to them. Clarifying at an early stage is better than making assumptions that could result in bugs and miscommunication later.
Communication after finishing a task:	After a member finishes a task, they should inform the rest of the members to check the compatibility of the backend and the frontend of that task, and to allow other members to start working on tasks dependent on that task.

Table 7 - shows the team working agreement for sprint 2 and its details

The Definition of Done

- Finish the implementation of all user stories
- Finish the implementation of the mobile app including all of its pages and functionalities
- Finish the implementation of the administrator panel including all of its pages and functionalities
- Testing the whole product thoroughly making sure no major bug persists

Assumptions

1. 3.25 story points per day on average.
2. Productivity after Apr 15 (estimated) will follow the same general trend of the first half of the sprint (actual productivity).
3. If a user story had multiple tasks assigned to different team members, the story points were split accordingly.
4. In this sprint, developers are already familiar with the technologies and tools used.
5. Developers will be available during their working hours to answer or clarify any questions.
6. Team members will be working on the project until the completion of the system (none of them will have an unexpected circumstance that would force them to leave).

Tasks for Each User Story

Title	Task 1	Task 2	Task 3
4. Save multiple delivery addresses in an account	Designing the UI	Handling the logic of saving the info in the DB	
9. Adding product to shopping cart	Devise database to store shopping cart information	Implement POST /cart for adding items to cart	Implement add to shopping cart button
10. Viewing the shopping cart	Designing UI for viewing items in cart	Implement GET /cart for accessing cart information	
11. Modifying the shopping cart	Implement POST /cart for modifying existing products in cart	Using the POST endpoint to set update requests when the user changes cart information	
12. Viewing the best deals	Implement /home for best deals information	Integrate best deals section into the home view	
13. Viewing a customized list of items	Run statistics on the database to understand which products relate to each other	Implementing logic of recommendation curated to each customer	Assigning a tab in the home page to display the results
14. Finalizing orders	Designing the UI	Programming the interface methods to the backend	Implementing the logic of saving the results to the DB and notifying the admin
15. Access order history	Save previous orders into an archive database	Devise an archive collection for storing history orders	Design UI for displaying history orders and their detail information
16. Payment	Backend sends an external payment link to frontend	Frontend jumps to third-party payment tools and report back the status of payment	
19. Out-of-stock notification	Check stock after every purchase	Notify Administrator	Unit test
20. Create offers and deals	Add offer field to products	implement the backend logic to changing siad fields	Unit test

21. View transactions	Return a list of saved transaction to user	Unit test	
22. Contacting customers	Query the database for subscribed users	Query the database for user's email	Send emails
23. Backup and restore data	Find recommended backup frequency	Setup the backup protocol in the hosting site	Test the sizes and legitimacy of backups
24. Print invoices and packing lists	Backend auto-generates the invoices and packing lists	Admin website provide entry for viewing and printing the documents	
25. Access to sales statistics	Aggregate orders and sales and find best sellers, units sold, and other needed information	Separate statistics based on categories, time, products.	Implement the logic needed for the admin to access the different statistics
26. Subscribing to alerts and newsletters	Admin website integrates third-party newsletter services	Adding the option to withdraw/sign up to the newsletter in the app	Sending push notifications in the app in case of alerts
27. Availability of out-of-stock items	Backend monitors the stock of items		
28. SSL Transactions	Set up SSL communication in the backend	Test for major SSL vulnerabilities	Implementing the protocol in the app
29. Frequently Bought Together	Aggregate items purchased in the same cart into each product after every purchase	Unit test	Integrating it in the product page
30. Contacting the administrator	Design UI for viewing contact information	Integrates chat function for contacting admin	
31. Return the order	Devise database to handle returned orders	Backend provides APIs for marking an order returned	Frontend design button for returning an order
33. Testing for Security Vulnerabilities of the Entire system	Testing for SQL injection	Testing for robustness against DDoS	

Table 8 - shows the user stories in Sprint 2 Backlog, and the tasks each story is divided into

Daily Sprint 2 Report

The daily sprint report details each day, what stories did the members work on and how many story points they finished on that day. In addition, it also shows their productivity on that day, where 100% means finishing 1 point on a given day. The notes show and overview of what happened on that day.The table is color coded as follows:

Shows that the person was over-productive, achieving 100% or higher productivity on that day, the team burnt 4 or more points, or someone finished a task (in the notes)

Shows that the person was under-productive, achieving 60% or less productivity on that day or burnt less than 3

Shows that the person is done with the points for that story, or that a person is done with the sprint (in the notes)

Shows that the person took a vacation on that day

Date	Omar				Phillip							Zayd								Zyad								Burnt Points
	20	24	27	33	9	14	15	16	28	26	31	13	19	21	22	23	25	29	31	4	9	10	11	12	14	30	31	
Story Points	2	1	1	8	2	1	1	8	3	3	2	8	1	1	1	3	2	2	1	1	3	2	1	1	2	2	2	
April 4 2021	0.9	0.0	0.0	0.0	1.4	0	0	0	0	0	0	1.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.2
April 5 2021	1.7	0.0	0.0	0.0	1.9	0.0	0.0	0.0	0.0	0.0	0.0	1.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	2.5
April 6 2021	2.0	0.4	0.0	0.0	2.0	1.0	0.3	0.0	0.0	0.0	0.0	2.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.6	0.0	0.0	0.0	0.0	0.0	0.0	4.5
April 7 2021	2.0	1.0	0.4	0.0	2.0	1.0	1.0	0.2	0.0	0.0	0.0	3.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	3.4
April 8 2021	2.0	1.0	1.0	0.6	2.0	1.0	1.0	1.4	0.0	0.0	0.0	5.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	2.7	0.0	0.0	0.0	0.0	0.0	0.0	4.5
April 11 2021	2.0	1.0	1.0	1.3	2.0	1.0	1.0	2.1	0.0	0.0	0.0	6.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	3.0	1.1	0.0	0.0	0.0	0.0	0.0	3.7
April 12 2021	2.0	1.0	1.0	1.9	2.0	1.0	1.0	3.5	0.0	0.0	0.0	7.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	3.0	1.8	0.0	0.0	0.0	0.0	0.0	4.0
April 13 2021	2.0	1.0	1.0	2.6	2.0	1.0	1.0	4.7	0.0	0.0	0.0	8.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	1.0	3.0	1.8	0.0	0.0	0.0	0.0	0.0	3.1
April 14 2021	2.0	1.0	1.0	3.4	2.0	1.0	1.0	5.8	0.0	0.0	0.0	8.0	1.0	0.7	0.0	0.0	0.0	0.0	0.0	1.0	3.0	2.0	0.8	0.0	0.0	0.0	0.0	3.8
April 15 2021	2.0	1.0	1.0	4.3	2.0	1.0	1.0	6.7	0.0	0.0	0.0	8.0	1.0	1.0	1.0	0.5	0.0	0.0	0.0	1.0	3.0	2.0	1.0	0.5	0.0	0.0	0.0	4.3
April 18 2021	2.0	1.0	1.0	5.0	2.0	1.0	1.0	7.7	0.0	0.0	0.0	8.0	1.0	1.0	1.0	1.2	0.0	0.0	0.0	1.0	3.0	2.0	1.0	1.0	0.6	0.0	0.0	3.5

April 19 2021	2.0	1.0	1.0	5.8	2.0	1.0	1.0	8.0	0.6	0.0	0.0	8.0	1.0	1.0	1.0	2.1	0.0	0.0	0.0	1.0	3.0	2.0	1.0	1.0	1.5	0.0	0.0	3.5
April 20 2021	2.0	1.0	1.0	6.3	2.0	1.0	1.0	8.0	1.6	0.0	0.0	8.0	1.0	1.0	1.0	2.6	0.0	0.0	0.0	1.0	3.0	2.0	1.0	1.0	2.0	0.3	0.0	2.8
April 21 2021	2.0	1.0	1.0	7.2	2.0	1.0	1.0	8.0	2.5	0.0	0.0	8.0	1.0	1.0	1.0	3.0	0.7	0.0	0.0	1.0	3.0	2.0	1.0	1.0	2.0	0.3	0.0	2.9
April 22 2021	2.0	1.0	1.0	7.5	2.0	1.0	1.0	8.0	3.0	0.3	0.0	8.0	1.0	1.0	1.0	3.0	2.0	0.1	0.0	1.0	3.0	2.0	1.0	1.0	2.0	0.9	0.0	3.1
April 25 2021	2.0	1.0	1.0	8.0	2.0	1.0	1.0	8.0	3.0	1.8	0.0	8.0	1.0	1.0	1.0	3.0	2.0	1.5	0.0	1.0	3.0	2.0	1.0	1.0	2.0	1.8	0.0	4.3
April 26 2021	2.0	1.0	1.0	8.0	2.0	1.0	1.0	8.0	3.0	2.9	0.0	8.0	1.0	1.0	1.0	3.0	2.0	2.0	0.7	1.0	3.0	2.0	1.0	1.0	2.0	2.0	1.0	3.5
April 27 2021	2.0	1.0	1.0	8.0	2.0	1.0	1.0	8.0	3.0	3.0	1.0	8.0	1.0	1.0	1.0	3.0	2.0	2.0	1.0	1.0	3.0	2.0	1.0	1.0	2.0	2.0	2.0	2.4
April 28 2021	2.0	1.0	1.0	8.0	2.0	1.0	1.0	8.0	3.0	3.0	2.0	8.0	1.0	1.0	1.0	3.0	2.0	2.0	1.0	1.0	3.0	2.0	1.0	1.0	2.0	2.0	2.0	1.0
April 29 2021	2.0	1.0	1.0	8.0	2.0	1.0	1.0	8.0	3.0	3.0	2.0	8.0	1.0	1.0	1.0	3.0	2.0	2.0	1.0	1.0	3.0	2.0	1.0	1.0	2.0	2.0	2.0	0.0

Table 9 - shows detailed report of sprint 1, including story points each person did,total points burnt on that day

Date	Productivity %				Notes
	Omar	Phillip	Zayd	Zyad	
April 4 2021	90	140	110	80	
April 5 2021	80	50	50	70	Zyad finished #4 and started #9
April 6 2021	70	130	130	110	Phillip finished #9,#14 and started #15 Omar finished #20 and started #24
April 7 2021	100	90	100	50	Phillip finished #15 and started #16 Zyad finished #4 and started #9 Omar finished #24 and started #27
April 8 2021	120	120	150	60	Omar finished #27 and started #33
April 11 2021	70	70	90	140	Zyad finished #9 and started #10
April 12 2021	60	140	130	70	

April 13 2021	70	120	120	0	Zyad took a vacation Zayd finished #13 and started #19
April 14 2021	80	110	90	100	Zayd finished #19 and started #21 Zyad finished #10 and started #11
April 15 2021	90	90	150	70	Zayd finished #21,#22 and started #23 Zyad finished #11 and started #12
April 18 2021	70	100	70	110	Zyad finished #12 and started #14
April 19 2021	80	90	90	90	Phillip finished #16 and started #28
April 20 2021	50	100	50	80	Zyad finished #14 and started #30
April 21 2021	90	90	110	0	Zyad took a vacation Zayd finished #23 and started #25
April 22 2021	30	80	140	60	Phillip finished #28 and started #26 Zayd finished #25 and started #29
April 25 2021	90	150	140	90	Omar done with Sprint 2
April 26 2021	70	110	120	120	Zayd finished #29 and started #31 Zyad finished #30 and started #31
April 27 2021	140	110	100	130	Zayd and Zyad done with Sprint 2 Phillip finished #28 and started #31
April 28 2021	80	150	120	70	Phillip and Everyone is done with Sprint 2
April 29 2021	0	0	0	0	

Table 10 - shows member's productivity and overview of the events of each day

Burndown Chart

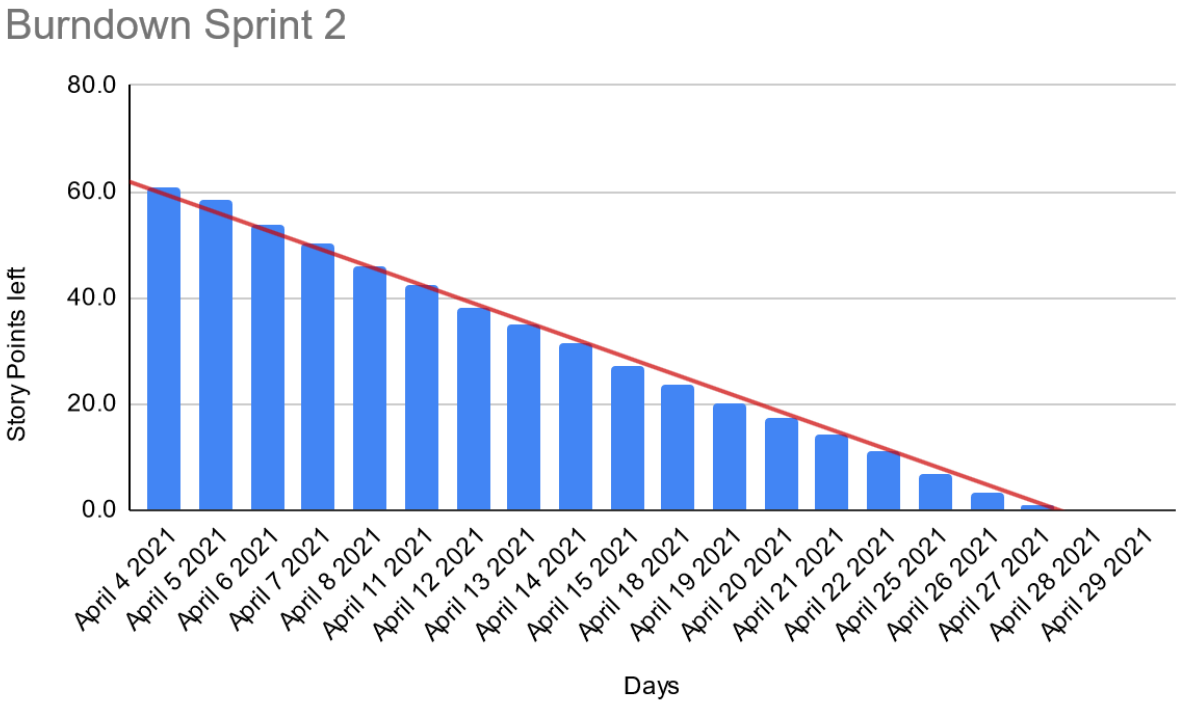


Chart 2. Shows the number of stories left per day during Sprint 2

Sprint Review

The team finished all the story points. Velocity= 65 points/sprint or 3.25 points / day

Product Burndown

We also compiled our work over the entire project to measure our velocity over the entire project and see trends in our efficiency over the project's duration.

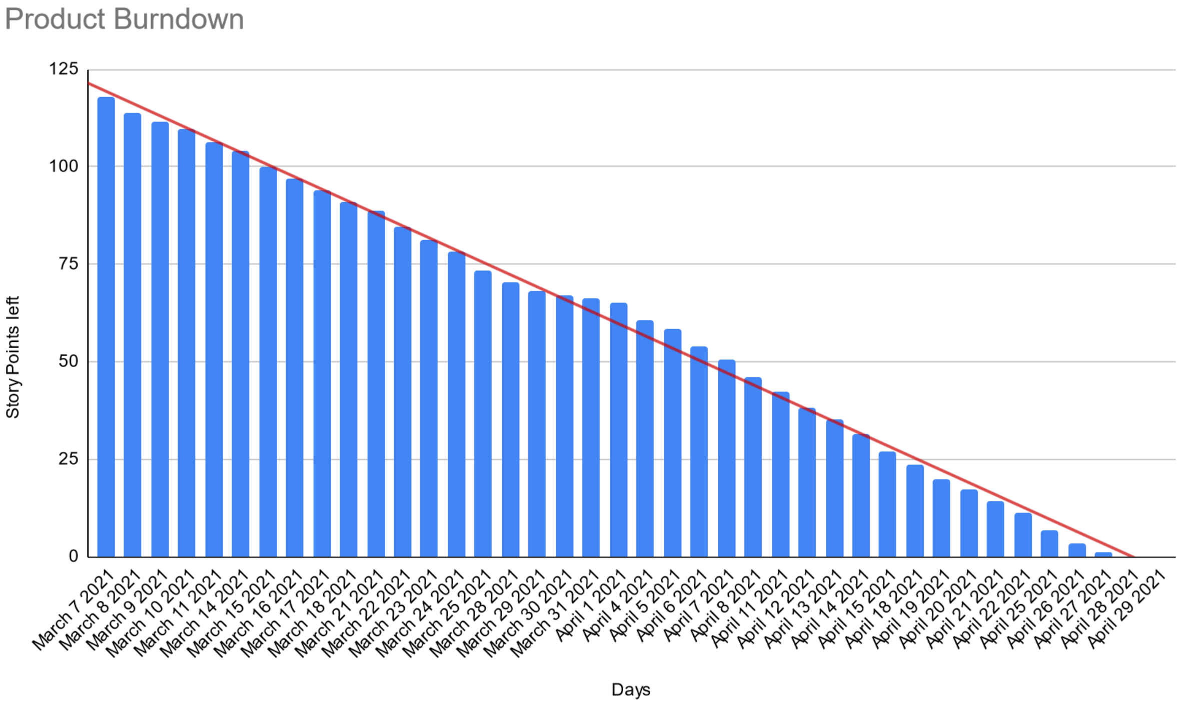


Chart 3. Shows the number of stories left per day during the whole development cycle

Velocity = 120 points / 40 days = 3 points per day or 120/2 sprints = 60 points per sprint.