

User Stories	Tasks	Description	Dependency	Story Points	1	2	3	4	5
1	1.1	Write our Tools Specifications that we will be using to build the platform. Documentation of technology we are using. (Sheet Team Plan)		4	R:2	R:2			
1	1.2	Create Heroku project and server.		2		B:2			
1	1.3	Set up Node.js backend on the Heroku server to show a simple web page containing HTML and CSS.	1.2	4			B:2	B:2	
1	1.4	Set up functionality to accept CSV files, by this we mean create the function that will handle the click request		2					B:2
2	2.1	Setup initial layout of web application, which involves node libraries/gitignores and standard files		2	D:2				
2	2.2	Create a simple HTML page with CSS styling (all with Bootstrap) which has a clickable upload button, with the X logo	2.1	10	C:2	C:2	C:2	C:2	C:2
2	2.3	Create a simple HTML page with CSS styled (all with Bootstrap) with Login Page.	2.1	10	A:2	A:2	A:2	A:2	A:2
2	2.4	Create a simple HTML page with CSS styled (all with Bootstrap) with Create Account Page.	2.1	5					
2	2.5	Create a Navigation Bar Sticky Header at the top of the webpage	2.1	5					
3	3.1	Flesh out the System Design of the Database Schema		8		D:2	D:2	D:2	D:2
3	3.2	Code Insert Report EndPoint (Accepts JSON Objects) (No DB Needed)		2	B:2				
3	3.3	Parse the json object element by element and insert it into the database	3.2	14			R:2	R:2	R:2
3	3.4	Use Axios to make a post request to the endpoints to send data.	3.2	8					
3	3.5	Read and parse data from an iCARE template, by using the npm package that converts csv to json	1.4	8					
		PLAN SPRINT # 1							

[illegible]

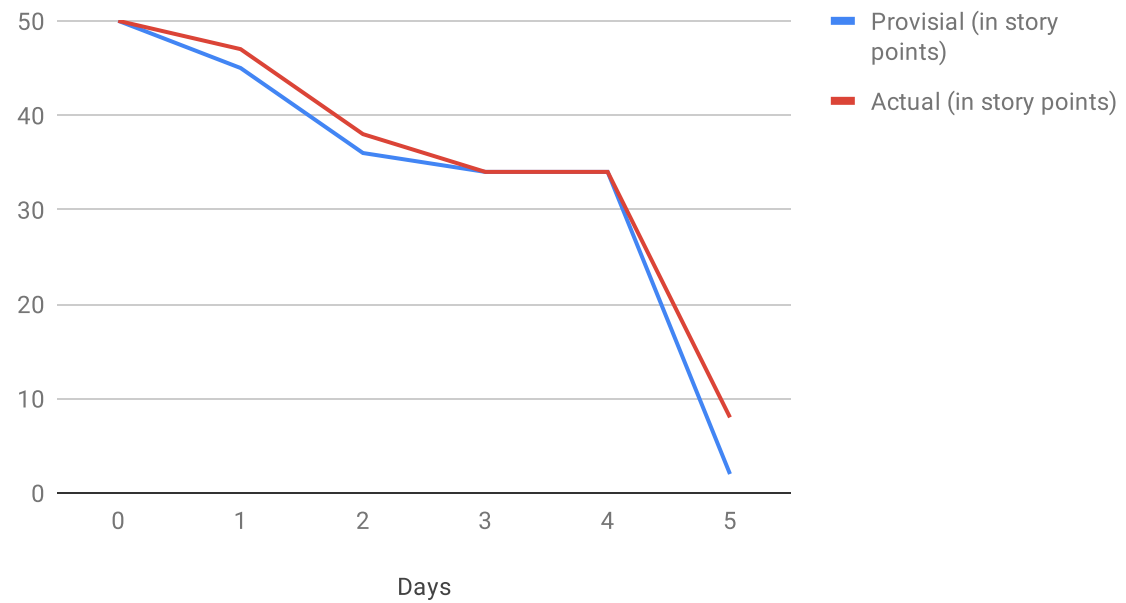


User Stories	Tasks	Description	Dependency	Story Points	1	2	3	4	5
1	1.4	Set up functionality to accept CSV files, by this we mean create the function that will handle the click request (be able to upload)		2	B:2				
2	2.5	Create a Navigation Bar Sticky Header at the top of the webpage		1	C:1				
2	2.6	Create the landing page and hook it up to the navbar		3	C:1	C:2			
2	2.7	Create the about us page and hook it up to the navbar		4	A:2	A:2			
2	2.8	Create a page that contains history of events that the TEQ can see. Make the UI only. In future we will get the data from the database.		4				C:2	C:2
2	2.9	Create a logout button for the page that the organization member goes to after they login, the local storage should reflect that they log out		2			C:2		
2	2.10	Create a Grid using an NPM Package for the list of people in the organization		6			A:2	A:2	A:2
3	3.3	Parse the json object element by element and insert it into the database		8	R:0	R:2	R:2	R:2	R:2
3	3.4	Read and parse data from an iCARE template, by using the npm package that converts csv to json	1.4	8		B:2	B:2	B:2	B:2
3	3.5	Send Axios to make post request when logging into account with local storage setup		2	D:2				
3	3.6	Send Axios to make post request when creating organization account		2		D:2			
3	3.7	Send Axios to make post request when uploading file data		6			D:2	D:2	D:2
		<b>PLAN SPRINT # 2</b>							



Days	0	1	2	3	4	5
Provisial (in story points)	50	45	36	34	34	2
Actual (in story points)	50	47	38	34	34	8

## BURNDOWN SPRINT # 2



	React + SC	JavaScript + CSS + HTML	
Aliza	Styled Components	CSS	
Bryon	JavaScript Front-End	Javascript	
Caleb	Styled Components	CSS	
Dharmik	JavaScript Front-End	JavaScript	
Ralph	BackEnd + DataBase ( Node Express and NEDB/MongoDB)	Node/Express/MongoDB	
Sprint #1	Sprint #2		
Wednesday - Sunday	Mon - Fri		
<b>Git Tutorial</b>			
Git Version Control	<a href="https://try.github.io/">https://try.github.io/</a>		
<b>How to Build &amp; Style a Web Page</b>		Caleb and Aliza google how to format a PDF with CSS	
BootStrap (CSS Styling)	<a href="https://getbootstrap.com/">https://getbootstrap.com/</a>		
Using HTML and CSS, bootstrap	<a href="https://www.w3schools.com/html/html_css.asp">https://www.w3schools.com/html/html_css.asp</a>		
Jquery (Making Web Dev Faster)	<a href="https://jquery.com/">https://jquery.com/</a>		
<b>How to talk to back-end from FrontEnd</b>			
JSON OBJECTS (How data Goes over the web)	<a href="https://www.json.org/">https://www.json.org/</a>		
Fetch or Axios Networking Library for Frontend	<a href="https://www.npmjs.com/package/axios">https://www.npmjs.com/package/axios</a>		
<b>How backend receives information from Front End</b>			
<a href="#">Node.js</a>	<a href="https://nodejs.org/en/">https://nodejs.org/en/</a>		
<a href="#">Express.js</a>	<a href="https://expressjs.com/">https://expressjs.com/</a>	app.get('/', (req, res) => res.send("Hello World!"))	This endpoint is what the front end people will hit. So LocalHost:/ will return Hello World when hit with a GET request
<b>How to Store Data</b>			
NEDB DataBase Technology	<a href="https://github.com/louischatriot/nedb">https://github.com/louischatriot/nedb</a>		
<b>Where will our code be deployed</b>			
Heroku it's like a Web Server	<a href="https://www.heroku.com/">https://www.heroku.com/</a>	When you push to master you will see updates at the URL Dharmik's Website.com	Ralph should set up CD
<b>Packages</b>			
Grid Package	<a href="https://gijgo.com/grid/demos/bootstrap-4-table">https://gijgo.com/grid/demos/bootstrap-4-table</a>	<a href="https://www.npmjs.com/package/jquery-grid">https://www.npmjs.com/package/jquery-grid</a>	
Upload a CSV file, return a JavaScript Array	<a href="https://www.npmjs.com/package/csvtojson">https://www.npmjs.com/package/csvtojson</a>	This converts CSV Files to JSON Objects that we can send to the Server Automatically since it's a JSON Object	
	<i>/* csv file</i>	Converts to JSON Object	
	<i>a,b,c</i>	[	
	<i>1,2,3</i>	* {a:"1", b:"2", c:"3"},	
	<i>4,5,6</i>	* {a:"4", b:"5", c:"6"}	
	<i>*/</i>	*/	
<b>Helpful Links</b>			
Bryon's Task	<a href="https://stackoverflow.com/questions/4720343/loading-basic-html-in-node-js">https://stackoverflow.com/questions/4720343/loading-basic-html-in-node-js</a>		