

Trash-Track

Sprint 1 Review



CEN Group 44

Joseph Somerdin
Ashish Satyavarapu
Giovanni Cornejo
Eric Clayton

- Product Manager
- Scrum Master
- Development Team Member
- Development Team Member

High Level Overview of Trash-Track





Trash-Track is a waste control web application.

- View and add nearby trash and recycling bin locations
- Gain experience points and level up when adding a new bins
- Share profile with friends
- Keep track of favorite bin locations

Purpose

- In Gainesville, litter can be found on the side of roads and near sidewalks
- This negatively affects ecosystems in and around the Gainesville area
- Our goal is to provide an easy way to locate nearby waste and recycling bins.
- Making trash disposal more convenient can prevent littering

Product Backlog Overview



Product Backlog

Only Important Items

▼ Backlog (12 issues)		408	0	0	Create sprint
TT-15	Enforce 30 minutes before logging another location	55	TO DO	▼	👤
TT-14	User clicks a bin to get its coordinates and open a navigation app of their choice	144	TO DO	▼	👤
TT-25	Followed/following accounts (by clicking on number)	21	TO DO	▼	👤
TT-32	Search for other profiles	55	TO DO	▼	👤
TT-17	User feedback that they received experience	21	TO DO	▼	👤
TT-20	Invite friends with link	13	TO DO	▼	👤
TT-21	User gains levels from experience, value is stored	34	TO DO	▼	👤
TT-23	Level is seen on user profile, photo, and bio	5	TO DO	▼	👤
TT-24	See number of followers/following on user profile	21	TO DO	▼	👤
TT-26	Navigation to other sections and tools	13	TO DO	▼	👤
TT-28	Max 20 alphanumeric bio	5	TO DO	▼	👤
TT-29	Upload/Delete photo	21	TO DO	▼	👤
+ Create issue					

Velocity Completion_{and} Predicted Velocity



Velocity Completion

Insights TT Sprint 1 ▾



Sprint commitment



552 points

Target isn't set yet

▼ TT Sprint 1 13 Oct – 27 Oct (10 issues)

Finish all items by the end date.

0 0 552 Complete sprint ...

TT-11 Set up a MongoDB atlas database for users and coordinate points.	13	DONE ▾	
TT-8 Set up basic google oauth in ExpressJS with PassportJS and session middleware.	55	DONE ▾	
TT-27 Description of the application and its purpose	5	DONE ▾	
TT-7 Create svg markers for trash cans and recycle bins respectively.	34	DONE ▾	
TT-5 Research ways to store large amounts of coordinate pairs and search through them efficiently.	34	DONE ▾	
TT-6 Work on visual prototype for the app (basic design).	144	DONE ▾	
TT-9 Reading through javascript.info and the documentation for svelte at svelte.com/docs	89	DONE ▾	
TT-10 Create logo for trash-track	89	DONE ▾	
TT-34 Include routes to all necessary pages	55	DONE ▾	
TT-33 Sign up/sign, user profile, settings buttons, etc.	34	DONE ▾	

+ Create issue

- 552 story points completed during our first sprint (34% of total story points completed).
- 1082 story points left for the next two sprints.
- First sprint was slow because we were learning new technologies.



Predicted Velocity

- We expect to complete 540 story points on average for each of the next two sprints.
- During Sprint 2, we plan to complete 674 story points.
- During Sprint 3, we plan to complete 408 story points.
- Will probably exceed both of these estimates after adding items to the backlog.

Sprint 2 Backlog

▼ TT Sprint 2 25 Oct – 15 Nov (7 issues)

Finish all items by the end of the sprint

674

0

0

Start sprint



- | | | | |
|-------------------------------------------------------------------------------------------------------|-----|---------|--|
| TT-19 Delete account | 1 | TO DO ▼ | |
| TT-22 Create user schema verification for database | 21 | TO DO ▼ | |
| TT-12 Get GPS coordinates from user | 34 | TO DO ▼ | |
| TT-35 User cannot upload coordinates of an existing bin; five meter radius checked around coordinates | 233 | TO DO ▼ | |
| TT-30 Enforce sign-in exceptions (DoB, Email, Username) | 8 | TO DO ▼ | |
| TT-13 Display closest five bins on a map | 233 | TO DO ▼ | |
| TT-36 Upload trash/recycling with coordinates input | 144 | TO DO ▼ | |

+ Create issue

Review of Test Driven Development





How we implemented Test Driven Development

- Jest to test lower level javascript utility functions.
- Insomnia to test higher level api http requests.
- Wrote tests first.

Jest

```
> server@1.0.0 test /home/joey/Projects/11282group44/server
> jest
```

```
PASS ./util.test.js
```

```
✓ distance between century tower and reitz union to equal ~0.005143 (1 ms)
✓ distance between century tower and southwest rec to equal ~0.027470
✓ distance between southwest rec and marston science library to equal ~0.0
26833
✓ distance between reitz union and southwest rec to equal ~0.022339
✓ distance between century tower and marston science library to equal ~0.0
00806
✓ closest point to 20, 82 to equal {lat: 29.6488, lng: -82.3433} (3 ms)
✓ closest point to 10, -82 to equal {lat: 29.6381, lng: -82.3686} (1 ms)
✓ closest point to 29.6379, -82.3690 to equal {lat: 29.6488, lng: -82.3433
}
```

```
Test Suites: 1 passed, 1 total
```

```
Tests: 8 passed, 8 total
```

```
Snapshots: 0 total
```

```
Time: 0.355 s, estimated 1 s
```

```
Ran all test suites.
```

```
~/Projects/11282group44/server · (joseph-somerdin±)
```

```
> █
```

Insomnia

(API Testing)

Application Edit View Window Tools Help

Dashboard / trash-track-test ▾

DESIGN DEBUG TEST

Setup Git Sync

New Test Suite

API

New Test Run Tests ▶

Tests Passed 10/10

Nearest Can Returns 200	api/nearest / [GET] Valid Nearest Can ▾	Passed	Nearest Can Returns 200	209 ms
<pre>1 const response1 = await insomnia.send(); 2 expect(response1.status).toEqual(200);</pre>		Passed	Nearest Can Returns 200	44 ms
Nearest Can Returns 200	api/nearest / [GET] Valid Nearest Can ▾	Passed	Nearest Can Returns 200	34 ms
Nearest Can Returns 200	api/nearest / [GET] Valid Nearest Can ▾	Passed	Nearest Can Returns 200	32 ms
Nearest Can Returns 200	api/nearest / [GET] Valid Nearest Can ▾	Passed	Nearest Can Returns 200	36 ms
Nearest Can Returns 400 (Invalid Query)	api/nearest / [GET] Invalid Nearest Ca ▾	Passed	Nearest Can Returns 400 (Invalid Query)	33 ms
Nearest Can Returns 400 (Invalid Query)	api/nearest / [GET] Invalid Nearest Ca ▾	Passed	Nearest Can Returns 400 (Invalid Query)	31 ms
Nearest Can Returns 400 (Invalid Query)	api/nearest / [GET] Invalid Nearest Ca ▾	Passed	Nearest Can Returns 400 (Invalid Query)	31 ms
Nearest Can Returns 400 (Invalid Query)	api/nearest / [GET] Invalid Nearest Ca ▾	Passed	Nearest Can Returns 400 (Invalid Query)	38 ms
Nearest Can Returns 400 (Invalid Query)	api/nearest / [GET] Invalid Nearest Ca ▾			
Nearest Can Returns 400 (Invalid Query)	api/nearest / [GET] Invalid Nearest Ca ▾			
Nearest Can Returns 400 (Invalid Query)	api/nearest / [GET] Invalid Nearest Ca ▾			

Demo





Thank you!

