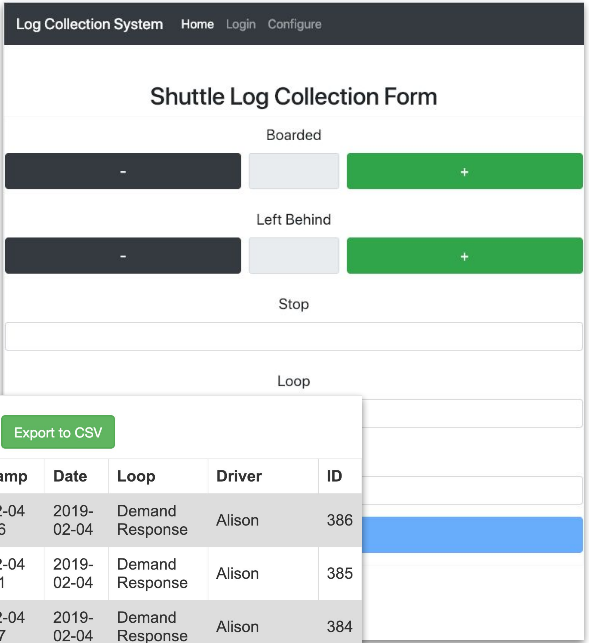
Bus Shuttle Log Collection System

Taneia Reed, Taylor Slusher, Keith DeSimini

Project Demo

- Driver Interface
- Desktop Interface - Displays data from yesterday, today, and all time.
- Automatic Retry
- User-Friendly Interface
- API
- Karma - Test Runner



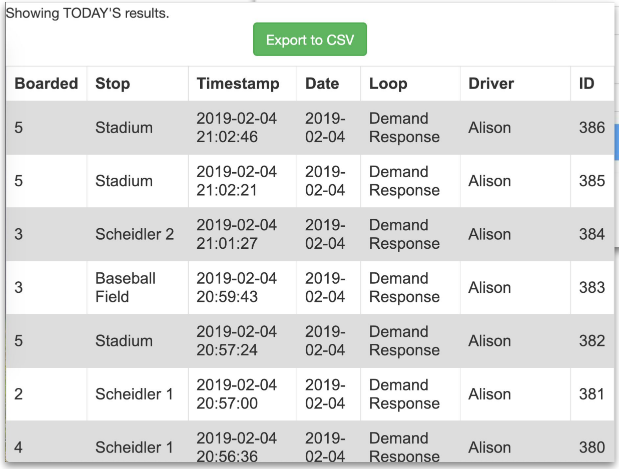
Shuttle Log Collection Form

Boarded

Left Behind

Stop

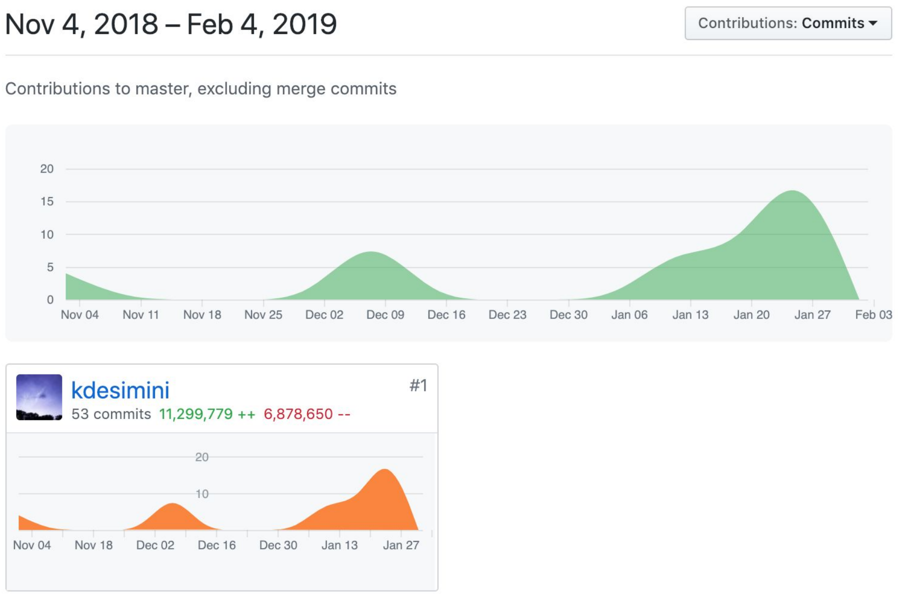
Loop



Showing TODAY'S results.

Boarded	Stop	Timestamp	Date	Loop	Driver	ID
5	Stadium	2019-02-04 21:02:46	2019-02-04	Demand Response	Alison	386
5	Stadium	2019-02-04 21:02:21	2019-02-04	Demand Response	Alison	385
3	Scheidler 2	2019-02-04 21:01:27	2019-02-04	Demand Response	Alison	384
3	Baseball Field	2019-02-04 20:59:43	2019-02-04	Demand Response	Alison	383
5	Stadium	2019-02-04 20:57:24	2019-02-04	Demand Response	Alison	382
2	Scheidler 1	2019-02-04 20:57:00	2019-02-04	Demand Response	Alison	381
4	Scheidler 1	2019-02-04 20:56:36	2019-02-04	Demand Response	Alison	380

Contributions



Nov 4, 2018 – Feb 4, 2019

Contributions: **Commits**

Contributions to master, excluding merge commits

kdesimini #1

53 commits 11,299,779 ++ 6,878,650 --

Requirements: Driver

Functional:

- Set the amount of people getting on the bus
- "Sign on" at the start of their shift
- Select the stop
- Be alerted if the the information was pushed or not
- Say if they left anyone behind
- Must work with sporadic internet connection

Non-Functional:

- Easy to use
- Not distracting
- Must be fast

Requirements: Desktop User

Functional:

- View the information from the drivers
- View the information as it is updating
- Filter information
- Export information

Desktop Admin User:

- Add/edit/delete buses/loops/stops.

Non-Functional:

- Easy to use
- The user interface must be be intuitive and easy to learn and use

Priorities

High Priorities:

- Working with the database as it is updating in as close to real time as possible
- View the daily information and sort by loop
- Exporting selected data into a form that can be opened with Excel

Mid Priority:

- Creating/Deleting/Editing stops/loops/routes

Low Priority:

- Sorting the data by week/month/year

Database

forms

formID	driverName	loopID	stopID	passengersOn	passengersLeft
int	char(25)	int	int.	int	boolean
primary key					

loops

loopID	loopName
int	char(25)
primary key	

Database continued

stops

stopsID	abreviation	stopName	stopDescription
int	char(10)	char(25)	varChar(125)
primary key			.

loops_stops

loopID	stopID
int	int
foreign key	foreign key

Database continued

- As of right now the database contains “test data” for forms.
- At one point the database contained entries for the stops and loops, however I was using an incompatible version of MYSQL with MYSQL Workbench. This meant that we were unable to export the database. I'm now using a compatible version, but the data has not been entered yet.
- The current version of the database is just a prototype, and we plan on improving it as we continue to study relational databases.