Fall 2018

Texas A&M University

CSCE 606

Software Engineering

TAMUber Safety Driver Interface

Submitted By:

Team Codebusters

Introduction

Stakeholders: Robert Brydia, Srikanth Saripalli, TTI

The TAMUber Safety interface is a web project created for the Texas A&M Transportation Institute's TAMUber project. This project is aimed at providing transportation assistance to disabled and mobility-disadvantaged students at Texas A&M. Our team worked on the Driver Safety interface for TAMUber project. The safety interface is geared towards ensuring the safety of passengers who will use these carts. We added multiple features in this application which will be useful for the driver for keeping the vehicle and passenger status.

To list out few items, we introduced a checklist in the application. This checklist will be mandatory for the driver to complete and the list contains items such as:

- Are all passengers wearing seat belts?
- Are the engine lights working?
- Does the car have enough cooling liquid for the day's trip?

If any of these items are not checked, the application doesn't allow the driver to enter the application. Additional to the checklist, we added a weather widget to the home page of the application. We build the login page and other features from scratch. We may consider using part of the admin's feature in the future. The customer's needs were for an interface that helps keep the passengers and the driver of the cart safe. Our website meets all their requirements. The safety drivers can log in to the vehicle's console. After logging in the driver has to check the checklist page before he can continue to the dashboard. The drivers can also keep track the weather to maintain the safety of the autonomous vehicles.

Finally, the driver will need a way to contact the authorities if an emergency occurs such as car accidents. For this, a calling button is provided that will immediately notify the authorities about the emergency via phone call.

Legacy Code:

Even though our project is a legacy project, there were little components that we could have reused in our application. Therefore, all the features in the application are built from scratch

Iterations

Iteration 0:

10/8 at 2:00 PM with Dr. Robert Brydia

Basic features were realized in this iteration. We discussed the holistic picture of the project and the customer requirements with Dr. Brydia. Two features were realized in this iteration

1. Login/Signup page for the vehicle
2. Dashboard for displaying vehicle condition/stats for the Safety Driver.

Iteration 1:

We did not get to meet with the customer physically, but we had email exchanges with the customer during this iteration.

Two more features were realized in this iteration

1. Add checklist for driver after logging into the app
2. Display Weather information for the driver

Iteration 2:

10/23 at 7:00 PM with Dr. Robert Brydia and Dr. Srikanth Saripalli

We discussed the common architecture and the data that is required for the project. We also proposed the modifications to the user-stories that we came up with and checked for the compliance.

We decided to add one more user story.

1. A calling button in the app in case of emergency.

Iteration 3:

We did not get to meet with the customer physically, but we had email exchanges with the customer during this iteration. We proposed a central server architecture for all the interfaces such as Safety, Student and Admin. Customer seemed to like the idea.

Iteration 4:

10/20 at 7:00 PM with Dr. Robert Brydia and Dr. Srikanth Saripalli

Features realized in from Iteration 1-3 were demoed:

The customer asked us to perform a few modifications.

1. Merge the weather page and vehicle dashboard on a single page
2. Display student information and their pick-up/drop-off locations on the same page

User Stories

Our web app has one stakeholder: Autonomous vehicle cart

1. Feature: **Sign Up**
   As a driver,
   I want to sign up to the Safety Interface,
   So that I can log in.

2. Feature: **Log In**
   As a driver,
   I want to log in to the Safety Interface,
   So that I can do operations in the system.

3. Feature: **Checklist for the driver**
   As a driver of the system
   So that I can make sure vehicle is in good working condition
   I want to check conditions of vehicle parts using a checklist

4. Feature: **Weather information for the driver**
   As a driver of the system
   I can view the weather conditions
   I want to be able to access it on the interface

5. Feature: **Real time updates on measurable parameters in vehicle**
   As a driver of the system,
   I should be able to monitor the car condition regularly from the sensors,
   If the vehicle has low tire pressure, LiDAR status etc.

6. Feature: **Emergency Call**
   As a driver of the system,

In case of emergencies,
I should be able to make a phone call from the interface

7. Feature: **Student Information**
   As a driver of the system,
   I should be able to view the student information onboard,
   I should also be able to see their trip details

For feature 6 we created Twilio Phone call MVC, routes. For the feature 3, the mockups are shown in Fig.1. Rest of the features mock-ups are shown in Fig 2. Screen shots are shown in rest of the figures below.
For checklist feature, we established the checklist page with pure front end coding and accomplished the following features:

- Modify the navbar to make sure that a driver cannot visit the dashboard page through the links on the navbar.
- Develop a checklist whose elements can fade away and be removed by clicking the check icon on each item.
- Create a button(link), which is disable until the driver have checked all items in the checklist.

System Design:

The design follows the MVC architectural pattern. This means that all interactions, including requests and business logic, are handled by the controller, which is the interface between the model(Data) and the view(UI logic). The controller used in our application is responsible for registering users, logging them in, and displaying information critical to the safety of the TamUber autonomous vehicle and its occupants.

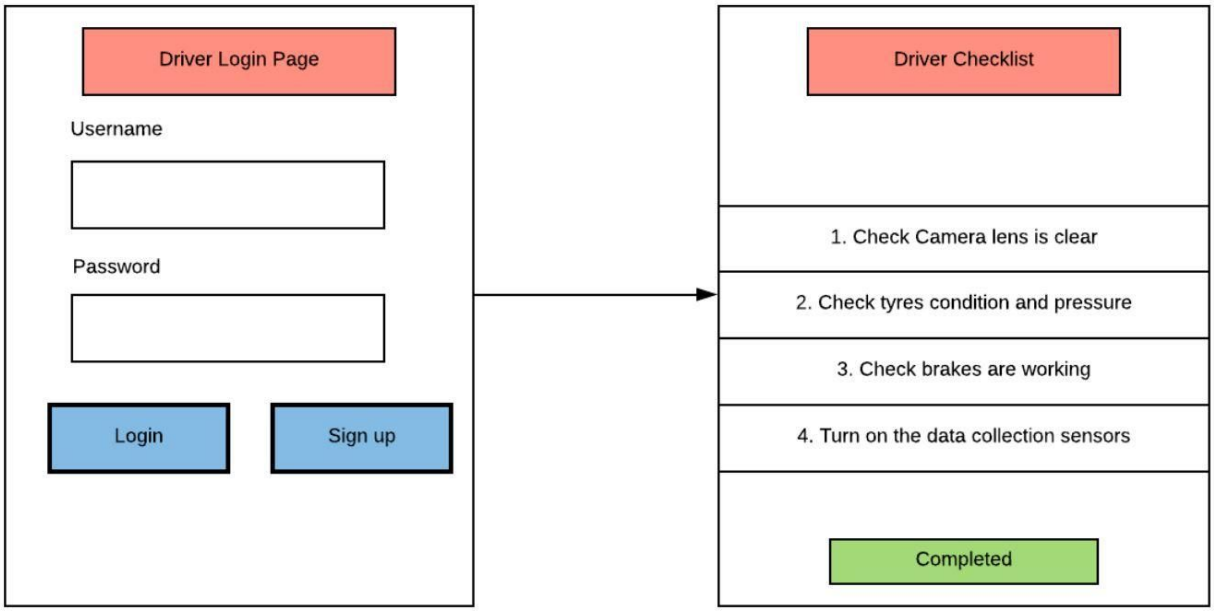


Fig 1. Checklist mockup

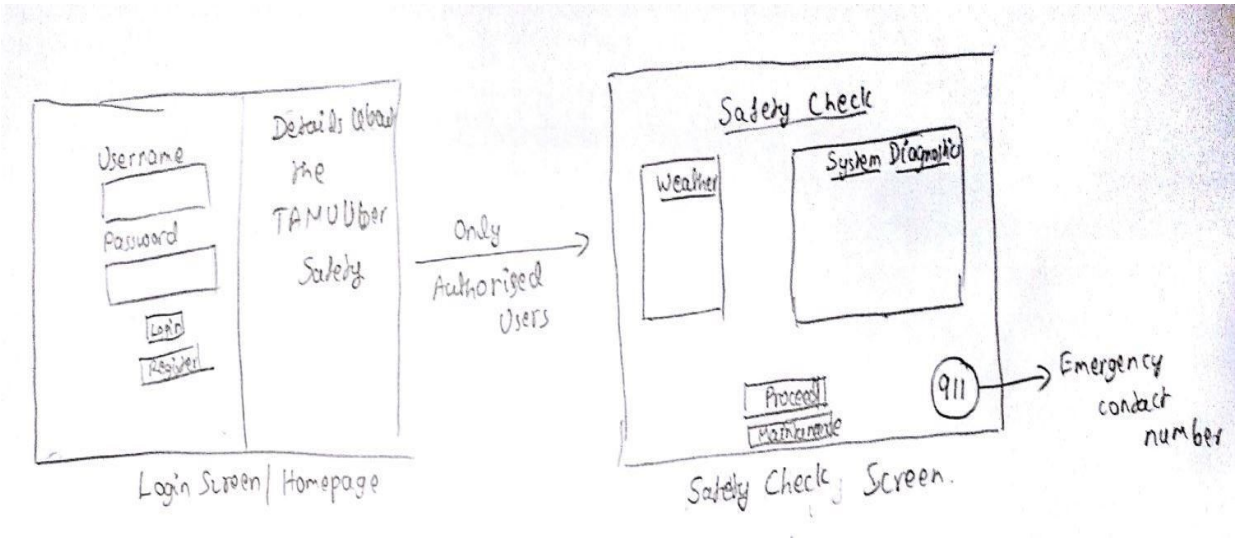


Fig 2. Mockup of Login/Signup/Calling Button and Dashboard

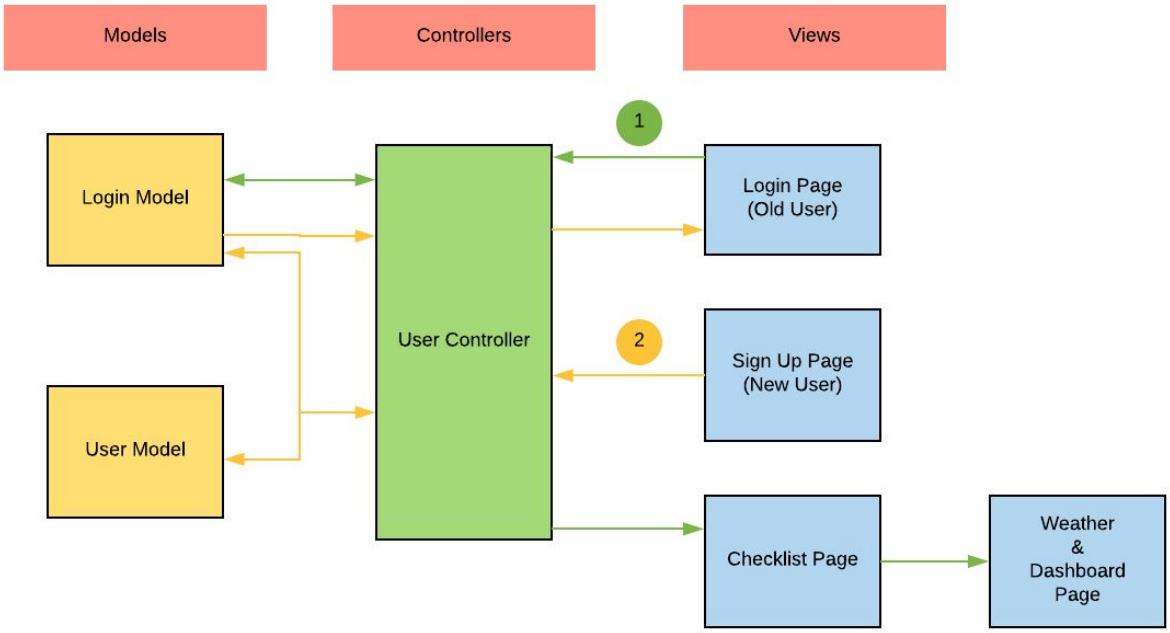
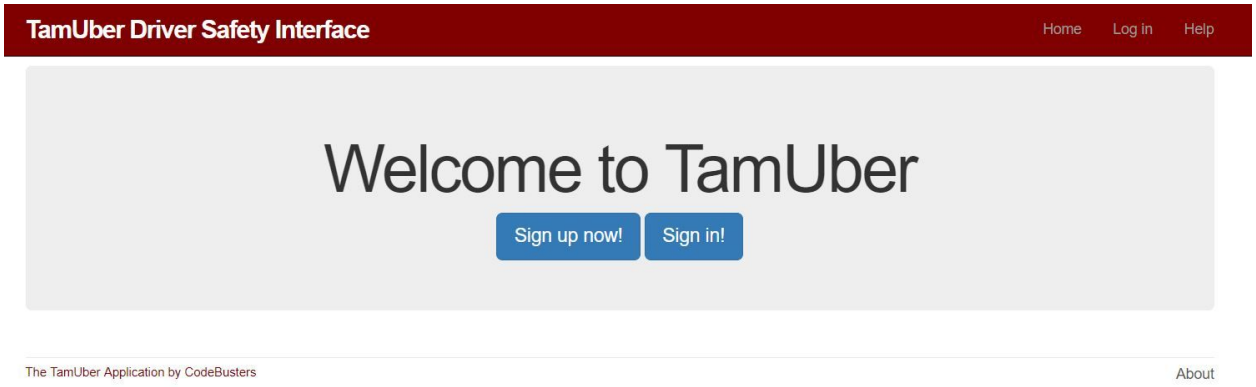


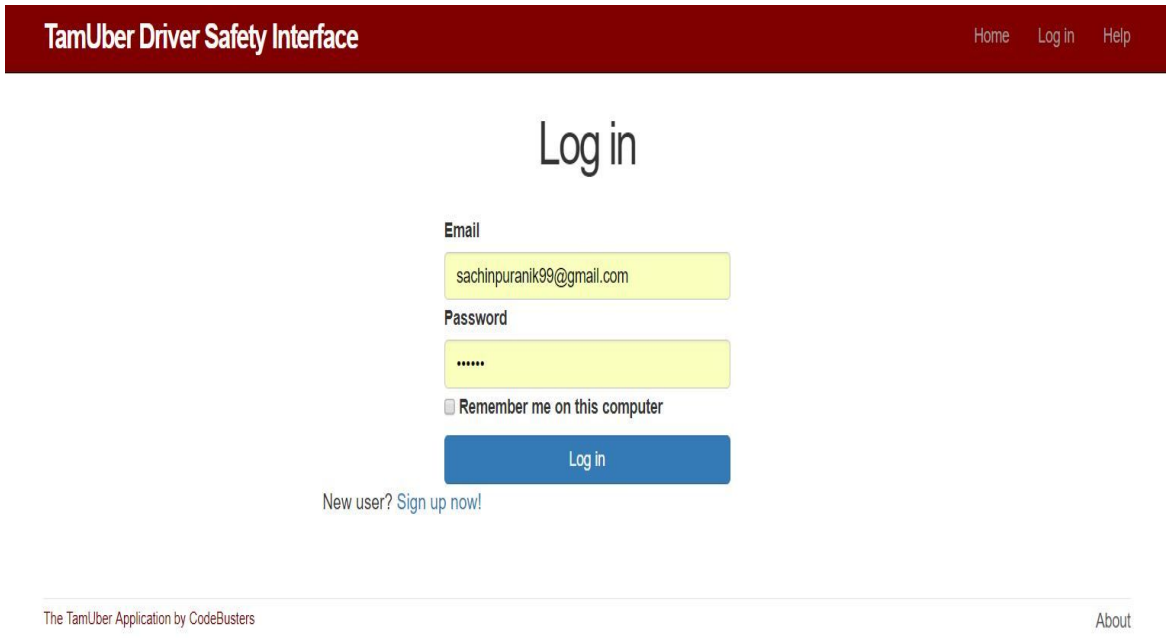
Fig 3. MVC of Student/Diagnostics

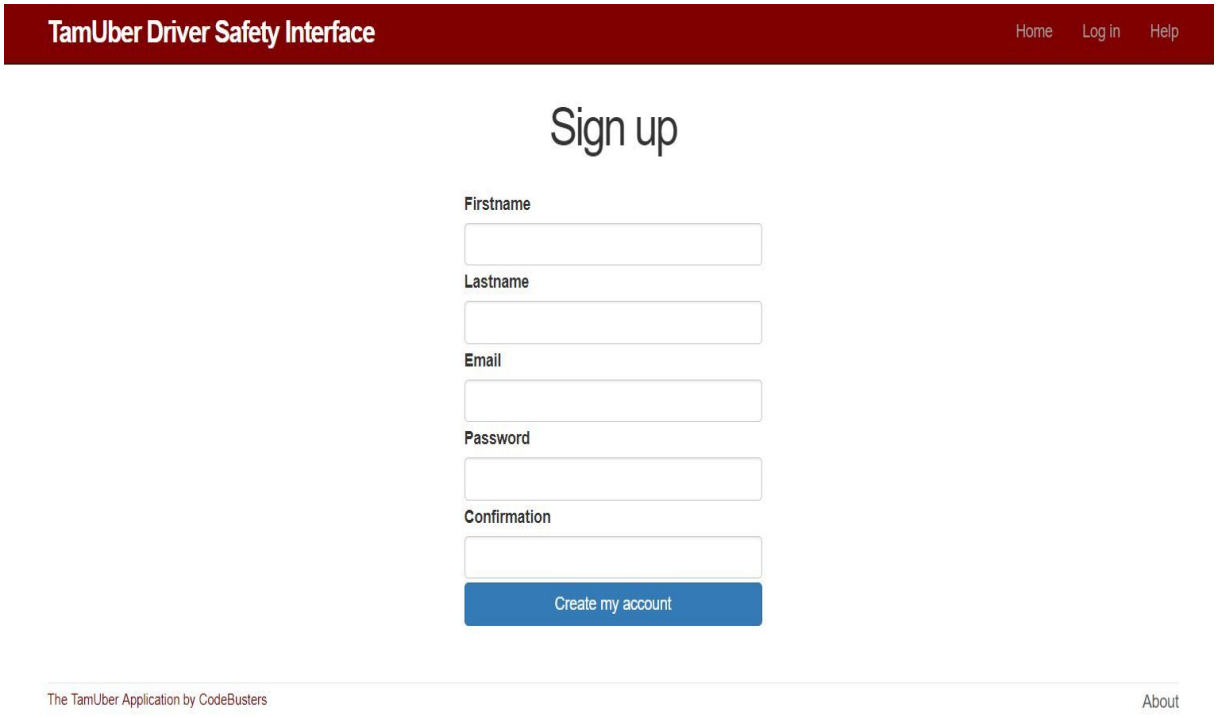
App Screenshots:

1. Landing Page:

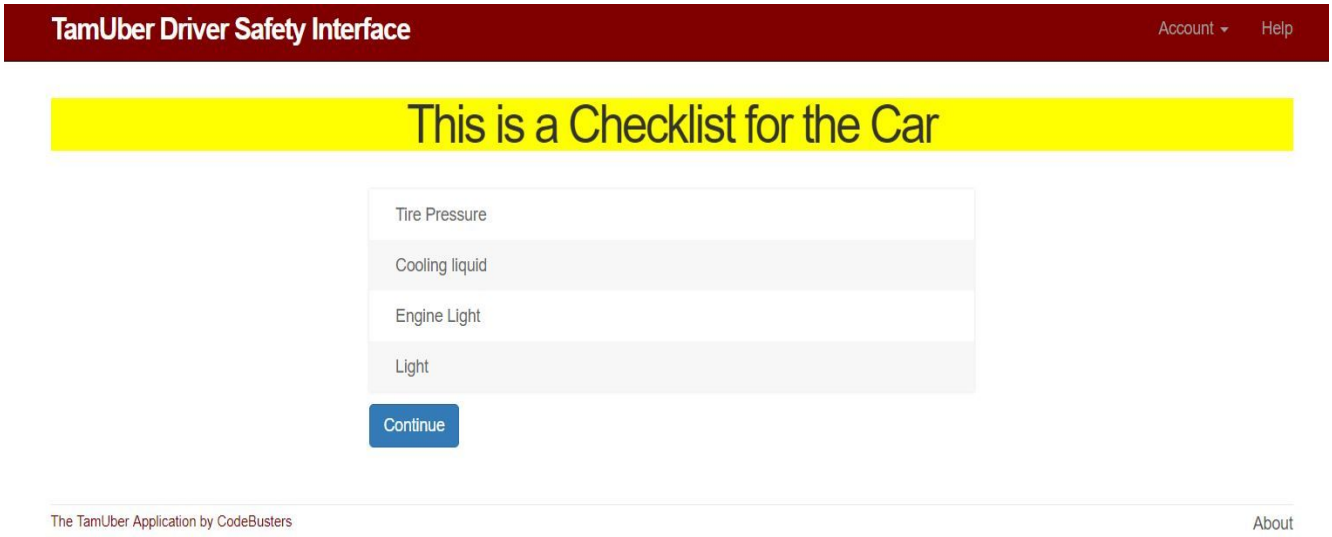


2. Sign-in and Sign-up page:

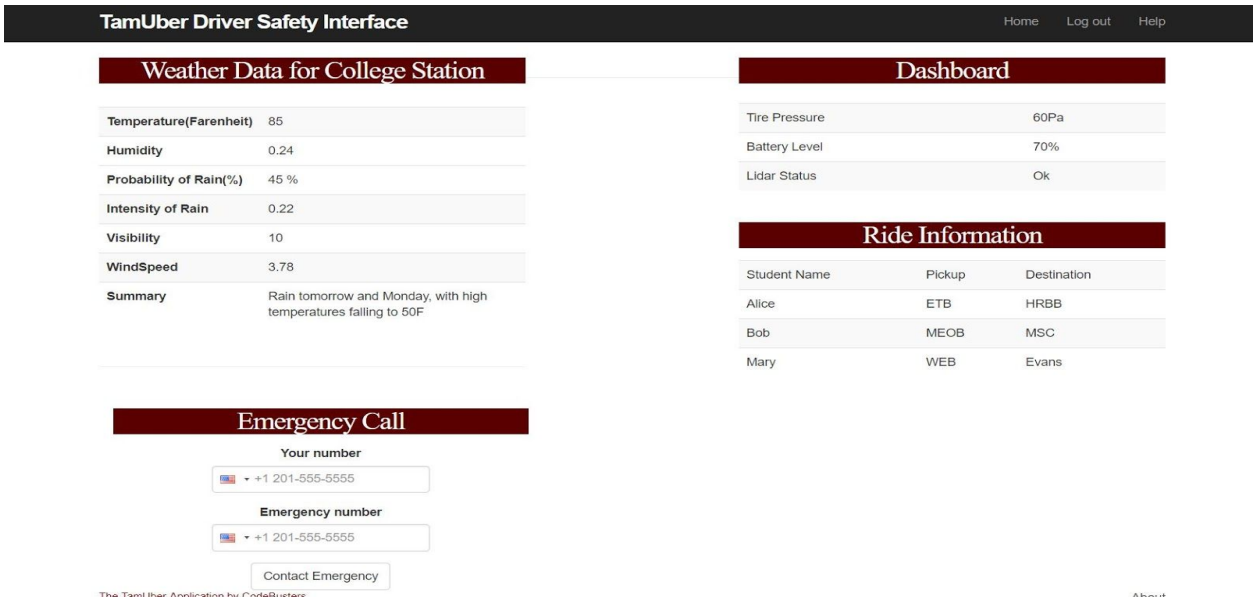

TamUber Driver Safety Interface
Home
Log in
Help
Log in
Email
sachinpuranik99@gmail.com
Password
Remember me on this computer
Log in
New user? Sign up now!
The TamUber Application by CodeBusters
About


TamUber Driver Safety Interface
Home
Log in
Help
Sign up
Firstname
Lastname
Email
Password
Confirmation
Create my account
The TamUber Application by CodeBusters
About

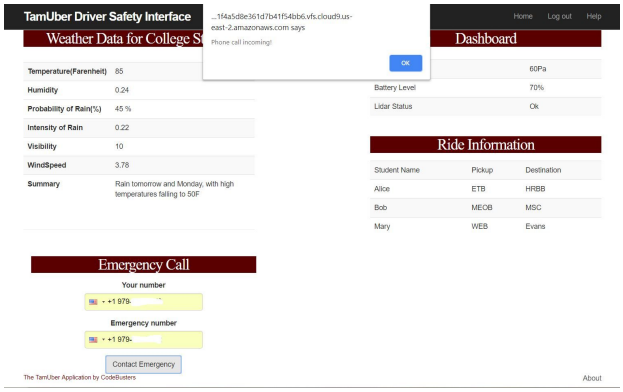
3. Checklist:

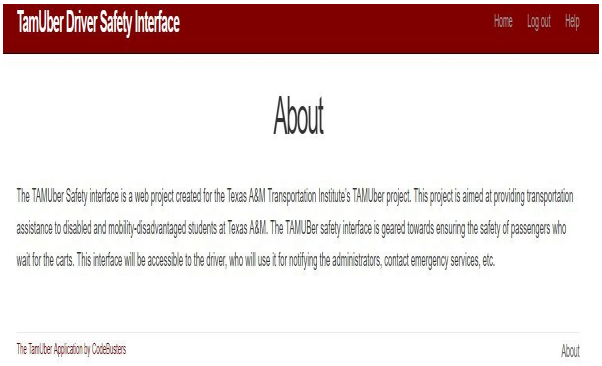


4. Dashboard:



5. Calling button and About page:





Project Execution

Communication Medium:

A WhatsApp group was created where all the six group members were added. This group was medium to communicate and discuss:

- Group meeting time
- Share individual status update with each other

Version Control Management:

We created a new branch every time when we created a new model. After we completed the changes in the feature branch, we switched to master branch and merged changes.

Project Challenges:

- Working with Heroku/Cloud9. One of the feature was working in Local environment but it was not working on the heroku side. Needed to add precompile steps to the production.rb to ensure that it was working on Heroku.
- Novice in Ruby on Rails framework
- Lack of effective coordination and communication between all the three teams, where the final product was combination of their products

Team Roles:

Roles	Name	UIN	Email
Product Owner/ Software Engineer	Akshay Murthy	326006338	akshay.ramuhally@tamu.edu
Scrum Master/ Software Engineer	Anjali Chadha	626008062	anjali_chadha@tamu.edu
Software Engineer	Kexin Cui	822001016	ckx9411sx@tamu.edu
Software Engineer	Sachin Puranik	427000135	pura247@tamu.edu
Software Engineer	Linxuan Wang	326006588	wanglx_2017@email.tamu.edu
Testing Engineer	Wenjie Zhang	427000298	wenjiezhang@tamu.edu

Table 1. Work Distribution about the final project

Useful GEMs:

- Gem 'bootstrap-sass': Use Bootstrap.js
- Gem 'phony_rails': Adds useful methods to your Rails app to validate, display and save phone numbers
- Gem 'twilio-ruby': Twilio calling REST API for phone calling features
- Gem 'pry-rails': Avoids repeating the initialization in a rails project.
- Gem 'database_cleaner': use Database Cleaner to clean data that created during testing.
  Gem 'jbuilder': Builds JSON APIs with ease
- Gem 'mocha': A Ruby library for mocking and stubbing.
- Gem 'jquery-rails': Provides the jQuery UJS adapter

Software Testing

BDD/TDD Process:

We considered the user stories and the scenarios first. Then we decided on how the code is implemented by TDD process. We benefit from BDD/TDD process in every iteration. BDD gives a correct direction to meet customer needs, while TDD guarantee a high quality of code.

Test Cases:

We built the test mainly using Cucumber and focused on test for 3 interfaces: signup, login and checklist. Some of the Test Cases against requirement is given below:

1. Ensure that new user can register by entering mandatory details in registration credential.
2. Ensure that error message is displayed when user leaves mandatory fields in the registration credential.
3. Ensure that error message is displayed when user provide invalid inputs in mandatory fields of registration credential.
4. Ensure that user can enter the site using login credential.
5. Ensure that error message is displayed when user enter the invalid username and valid Password in the login credential.
6. Ensure that error message is displayed when user leaves mandatory fields in the login credential.
7. Ensure that error message is displayed when user enter the valid username and invalid password in the login credential.
8. Ensure that User details are available when logged into the user account.
9. Ensure that system should not create more than one account for the same email address
10. Ensure that user can able to sign out from the site properly.
11. Ensure that user must check everything in checklist to continue.
12. Ensure that user can see the the weather information, dashboard for the vehicle data, the emergency and the ride information after clicking the continue button.

Test Coverage Screenshots:.

config/application.rb	100.0 %	18	6	6	0	1.0
config/boot.rb	100.0 %	3	2	2	0	1.0
config/environment.rb	100.0 %	5	2	2	0	1.0
config/environments/test.rb	100.0 %	42	12	12	0	1.0
config/initializers/application_controller_renderer.rb	100.0 %	8	0	0	0	0.0
config/initializers/assets.rb	100.0 %	14	2	2	0	1.0
config/initializers/backtrace_silencers.rb	100.0 %	7	0	0	0	0.0
config/initializers/cookies_serializer.rb	100.0 %	5	1	1	0	1.0
config/initializers/filter_parameter_logging.rb	100.0 %	4	1	1	0	1.0
config/initializers/inflections.rb	100.0 %	16	0	0	0	0.0
config/initializers/mime_types.rb	100.0 %	4	0	0	0	0.0
config/initializers/session_store.rb	100.0 %	3	1	1	0	1.0
config/initializers/wrap_parameters.rb	100.0 %	14	2	2	0	1.5
config/routes.rb	100.0 %	17	16	16	0	1.0
features/step_definitions/check_list_steps.rb	100.0 %	20	9	9	0	1.0
features/step_definitions/log_in_steps.rb	100.0 %	26	14	14	0	1.1
features/step_definitions/map_steps.rb	100.0 %	6	0	0	0	0.0
features/step_definitions/sign_up_steps.rb	100.0 %	48	25	25	0	1.1

Showing 1 to 29 of 29 entries

Figure 4 : First part of test coverage

All Files (74.5%) Generated less than a minute ago

All Files (74.5% covered at 1.41 hits/line)

29 files in total. 200 relevant lines. 149 lines covered and 51 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/sessions_controller.rb	30.77 %	24	13	4	9	0.3
app/helpers/users_helper.rb	33.33 %	10	6	2	4	0.3
app/helpers/sessions_helper.rb	40.0 %	46	25	10	15	4.0
app/controllers/users_controller.rb	44.44 %	37	18	8	10	0.4
app/helpers/application_helper.rb	50.0 %	23	12	6	6	3.5
app/models/user.rb	66.67 %	40	21	14	7	0.8
app/controllers/application_controller.rb	100.0 %	7	3	3	0	1.0
app/controllers/static_pages_controller.rb	100.0 %	14	5	5	0	1.0
app/helpers/bubble_helper.rb	100.0 %	2	1	1	0	1.0
app/helpers/static_pages_helper.rb	100.0 %	2	1	1	0	1.0
app/models/application_record.rb	100.0 %	3	2	2	0	1.0

Figure 5: second part of test coverage

Link Summary:

Customer Interview	https://vimeo.com/295515355
Video Demo	https://vimeo.com/305644668
Pivotal Tracker	https://www.pivotaltracker.com/n/projects/2200370
GitHub Account	sachinpuranik2007@gmail.com
GitHub Repo Link	https://github.com/team-codebusters/TamUber
Heroku	https://tamuber-safety.herokuapp.com/