Complete Document
Pullin' Freight

Young Jin Kim(kim687@usc.edu)

Jae Jun Min(jaejunmi@usc.edu)

Stakeholders: Marcy Pullard & Ken Pullard (Pullin' Freight LLC)

Table of Content

Project Description:

Operation Dashboard(Desktop Application):

The Operation Dashboard is for the stakeholder use only. Stakeholder uses this app to manage invoices, drivers' information and job tasks for each driver as well. Stakeholder is able to send jobs to each driver and once stakeholder sends the job to certain driver, driver can receive the job information from his/her mobile application. Stakeholder can also create the mobile application account for drivers by using this desktop application. Since the Pullin' Freight is private business, only stakeholder is allowed to create account for driver(s). With this application, stakeholder can extract the detail invoices and drivers' work information as excel files.

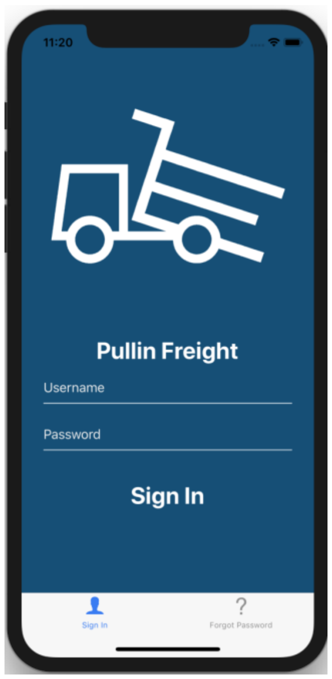
Mobile Application(iOS & Android):

The Pullin' Freight mobile application is only for the drivers who receive jobs from Pullin' Freight. This mobile application helps drivers to keep track of their job histories as well as current and future jobs. They can receive jobs from Pullin' Freight and decide whether they will accept or decline. Once they decide to take any jobs given by the stakeholders, they would able to look at the job information and fill out bill of lading once they successfully accomplished their tasks. Other than this main feature, there is authentication feature where user can change their information or check their important expirations dates that would be required in order to be part of Pullin' Freight's drivers. The purpose of this mobile application is to allow managers to keep track of drivers' information as well as their jobs.

Glossary

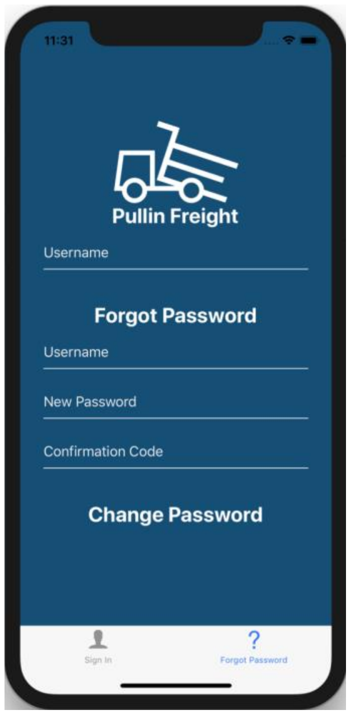
Database	Place to store all the transactions of all the users
Server	A central management that controls the flow of data between database and end users
UI	Short for “User Interface”, meaning the appearance of the app
GUI	Short for “Graphical User Interface”, is a form of user interface that allows user to interact with electronic devices through graphical icons
React Native	Outside library by Facebook that generates code for both iOS and Android platform at the same time
AWS	“Amazon Web Services”, a web computation service provided by Amazon
Executable File	File that is used to perform various functions or operations on a computer.
MySQL	An Open-source relational database management system. Used to create database schema.
PyInstaller	Puts python applications into standalone executables, under Windows, GNU/Linux, Mac OS.
Python	High level, General-purpose programming. Used to create the Operation Dashboard(Standalone) application.
PyQt5	Python binding of the cross-platform GUI toolkit. Used to create graphical interface for the Operation Dashboard(Standalone) Application

Functionalities



Mobile Application

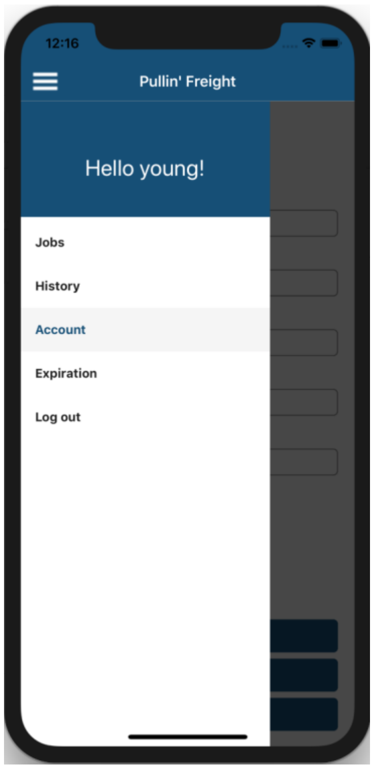
1. **Application Launch**
    a. This mobile application is not currently deployed in public. Currently, it is in development mode and can only be accessed through Xcode.
2. **Account Sign In**
    a. Once user launch this mobile application, the first page is sign in page. This application doesn't have sign up page because stakeholder doesn't want users to create their own account(Stakeholder want to sort their usernames in orders).
    b. User can log in once he/she received account information from stakeholder.
3. **Account Forgot Password**
    a. If user doesn't remember the password, the can change their password through this page. However, user need to verify their email once he/she logs in. This functionality only works when email is verified.
    b. When user want to reset password, he/she would insert user's username and press 'Forgot Password' button. This will send the verification code to user's verified email. Then, user need to enter the username, new password, and confirmation code. Lastly, once they press 'Change Password' button, the password will change.

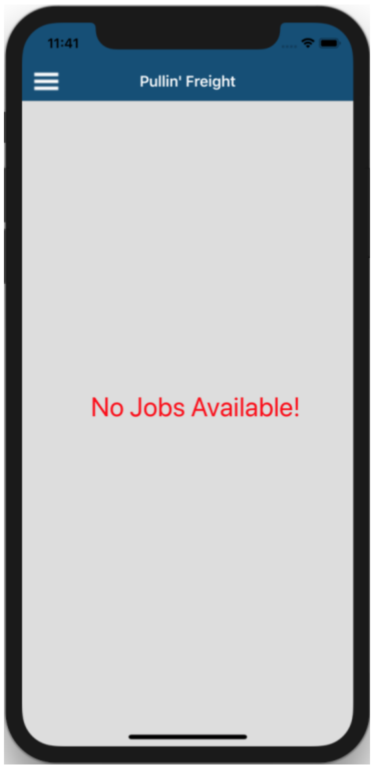


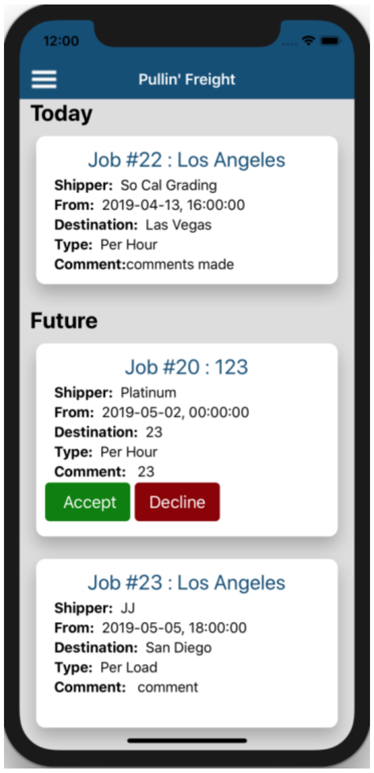
4. **Side Menu Bar**
    a. On the side menu bar there are five choices: Jobs, History, Account, Expiration, and Sign Out.
5. **Job Feed Page**
    a. Once user logs in, it will direct the user to job feed page which displays all jobs dispatched by operation dashboard. This will have jobs that stakeholder assigned them to which allow user to accept or decline by pressing either buttons.
    b. When user accepts pending job, it will remain on the job feed page and buttons for accept/decline will disappear. When user declines pending job, it will be removed from the job feed page.

c. Additional Features in this page are scrolling down job list to refresh or displaying “No Jobs Available!” text.

d. There are two sections: “Today” and “Future”. “Today” section will show the jobs that is for today and “Future” will display all other future jobs.







Today

Job #22 : Los Angeles
Shipper: So Cal Grading
From: 2019-04-13, 16:00:00
Destination: Las Vegas
Type: Per Hour
Comment:comments made

Future

Job #20 : 123
Shipper: Platinum
From: 2019-05-02, 00:00:00
Destination: 23
Type: Per Hour
Comment: 23

Job #23 : Los Angeles
Shipper: JJ
From: 2019-05-05, 18:00:00
Destination: San Diego
Type: Per Load
Comment: comment

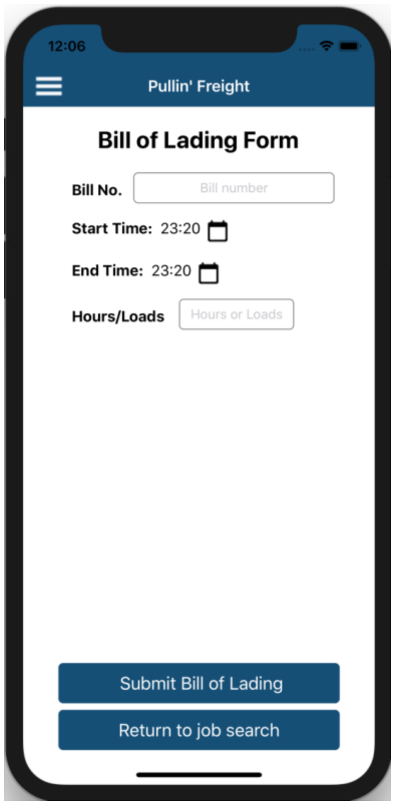
6. Bill of Lading Submission Page

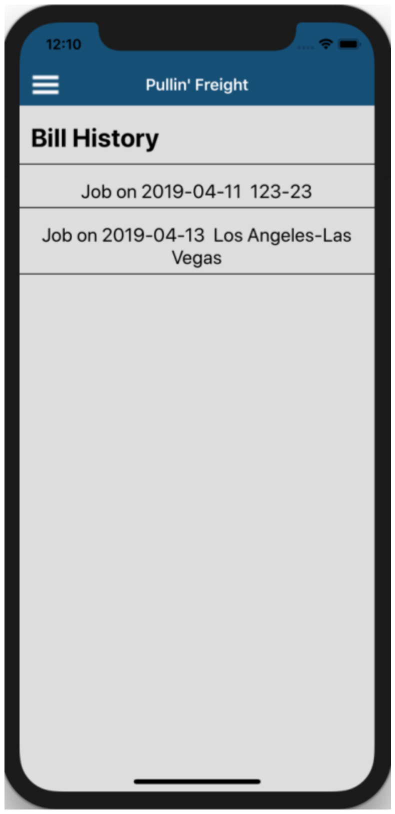
a. After user accepts the job, the user is responsible for completing accepted job.

b. After user completed accepted job, the user will fill out bill of lading to submit job’s invoice information by clicking the job that was completed and fill out the required fields.

c. "Submit Bill of Lading" will make bill of lading and send it to the database. Stakeholder will be able to check who filled out bill of lading through the dashboard. "Return to job search" go back to the Job Feed Page.

7. **Job History Page**
   a. In the side menu, the user can navigate to Job History Page where user can see all the page jobs' information based on bill of lading submission.
   b. Once user clicks on individual job listed in this page, it will display the information.
   c. Additional features in this page is refreshing list of jobs once they scroll down the job list and displaying "History is empty!" when user didn't finish any jobs

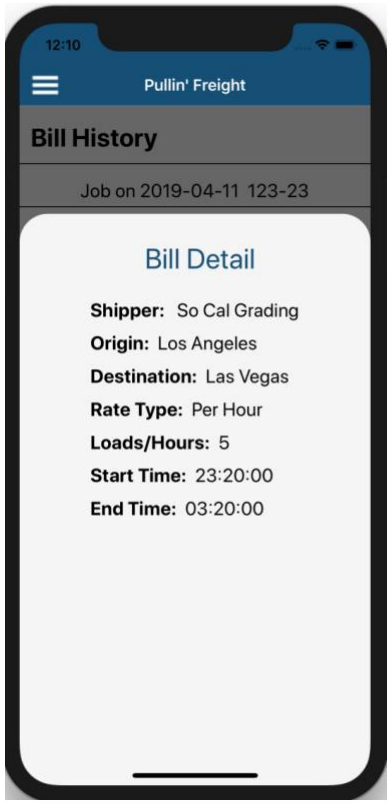




Bill History

Job on 2019-04-11 123-23

Job on 2019-04-13 Los Angeles-Las Vegas



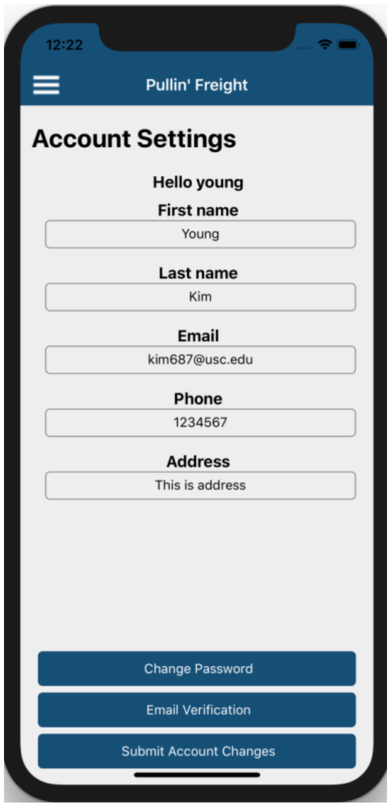
Bill Detail

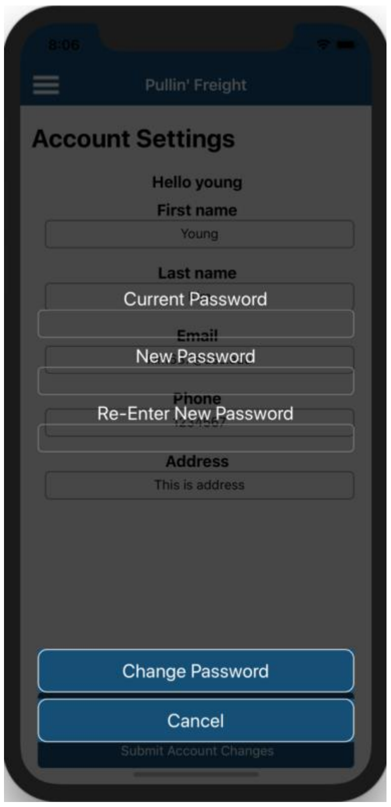
Shipper: So Cal Grading
Origin: Los Angeles
Destination: Las Vegas
Rate Type: Per Hour
Loads/Hours: 5
Start Time: 23:20:00
End Time: 03:20:00

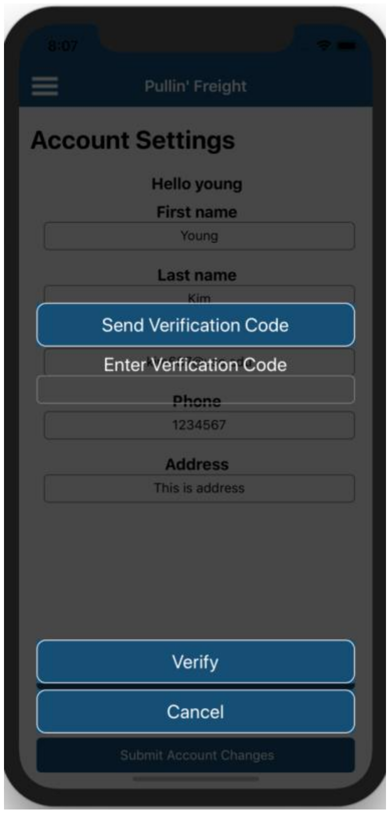
8. **Account Settings Page**
   a. Account settings page allow users to change their information once they get access from the stakeholder.
   b. There are three buttons on the bottom of this page. First button is to change current password. This connects to AWS Cognito and once they change from this screen, this will update AWS Cognito user password.
   c. Next button is Email verification screen. In order to use feature that can find account when they forgot the password, they need to have verified email in their AWS Cognito system. Since stakeholder will create individual's account, every account will be created without email verification so user need to use this screen

to verify email. “Verify Email” will send a verification code to the email. Then user need to use that code to verify.

d. Last buttons is to change the account settings. Once user clicks this button after account information has changed, it will update the new information into the database as well as AWS Cognito.


12:22
Pullin' Freight
Account Settings
Hello young
First name: Young
Last name: Kim
Email: kim687@usc.edu
Phone: 1234567
Address: This is address

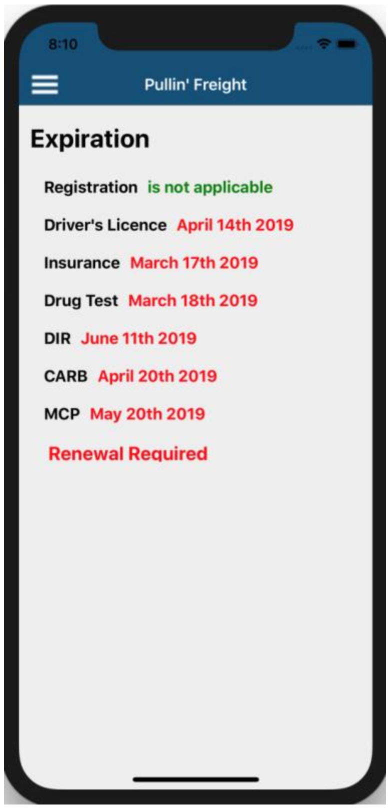




9. Expirations Page

a. This page displays all the list of expirations dates. If the expiration date is within forty-five days, it will show it in red text. Otherwise, it will show it on green text.

b. If at least one expiration date is in red text, it will say “Renewal Required”.



8:10

Pullin' Freight

Expiration

Registration is not applicable

Driver's Licence April 14th 2019

Insurance March 17th 2019

Drug Test March 18th 2019

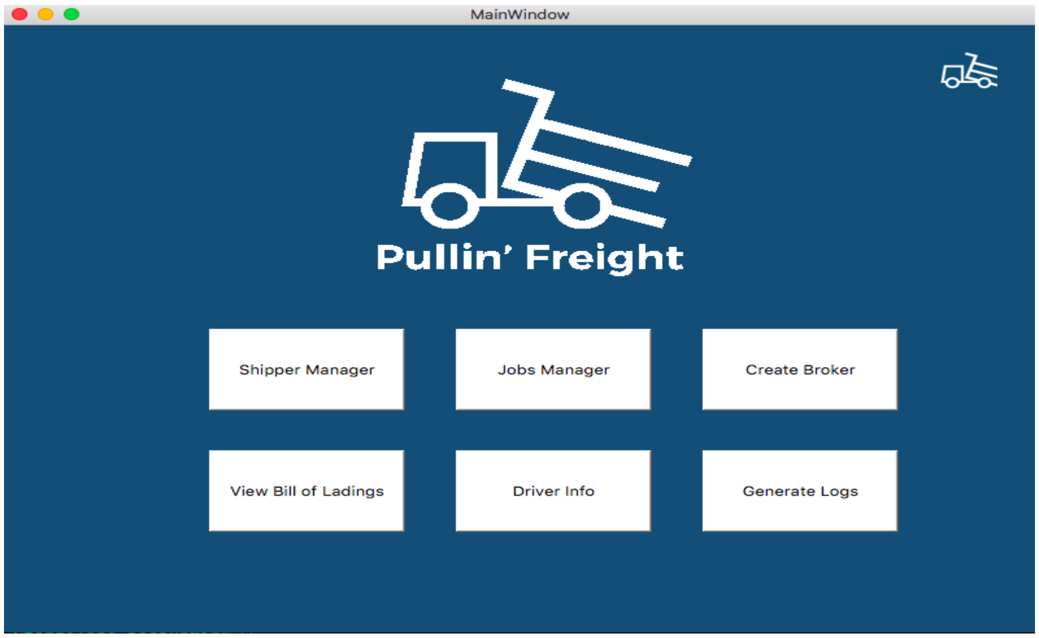
DIR June 11th 2019

CARB April 20th 2019

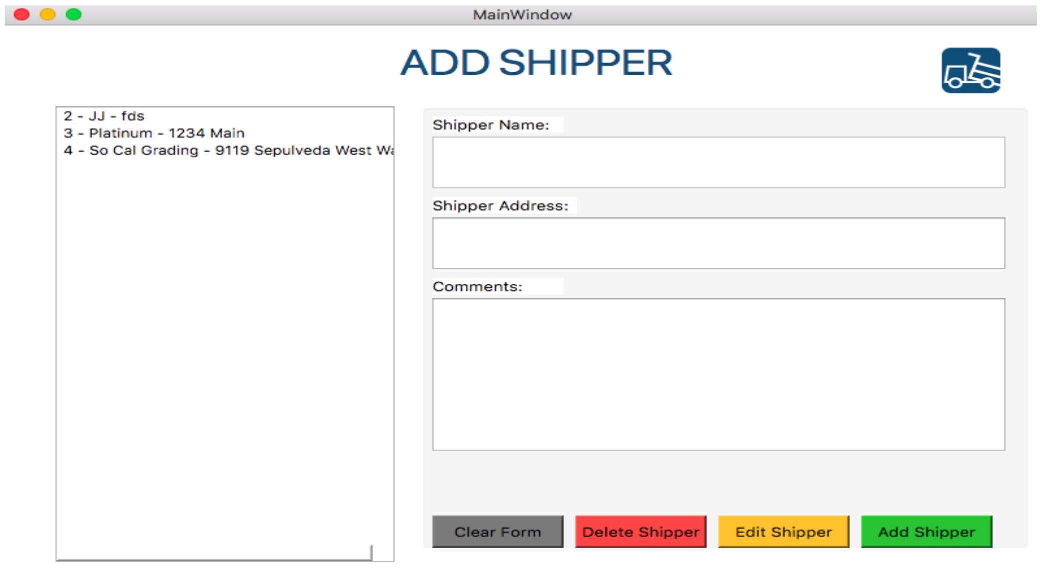
MCP May 20th 2019

Renewal Required

Operation Dashboard

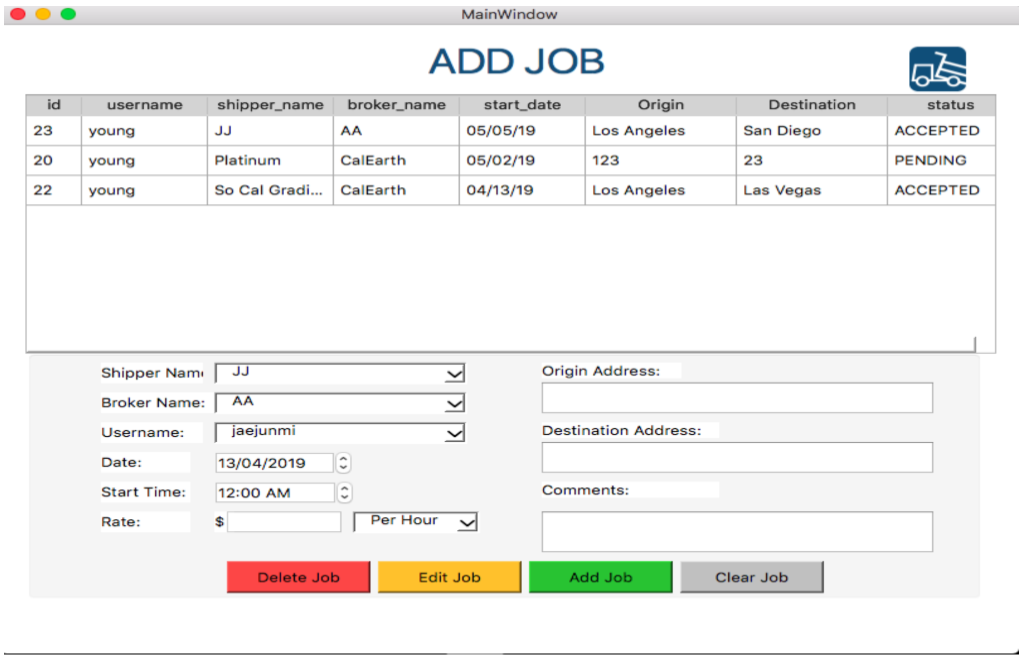


1. **Shipper Manager**
    a. User can create, edit and delete the shipper information.
    b. Display window on right side displays the list of shippers.



2 - JJ - fds
3 - Platinum - 1234 Main
4 - So Cal Grading - 9119 Sepulveda West W

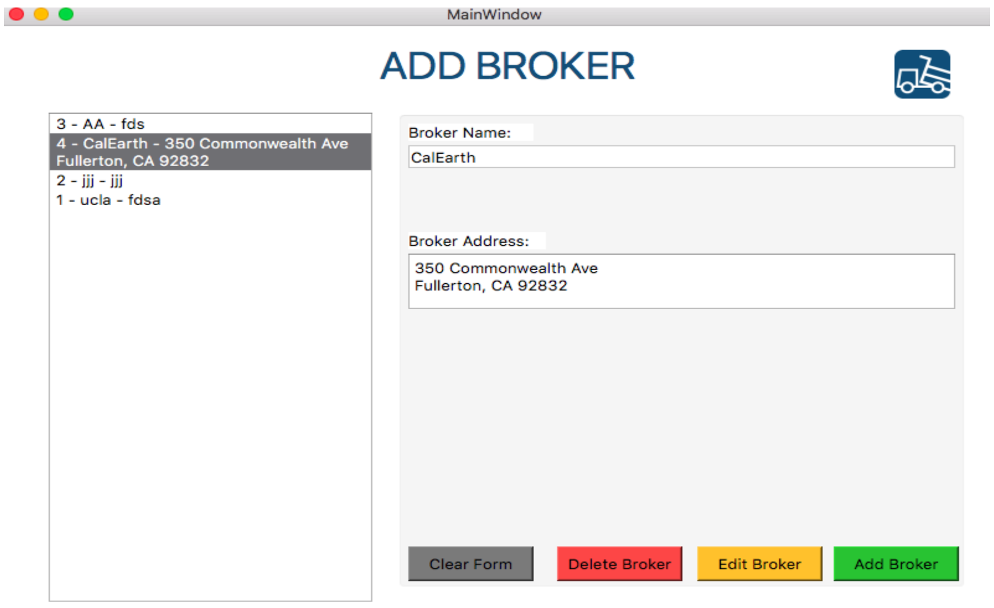
2. **Job Manager**
    a. Add Job
        i. User can assign driver a job with all the required informations (shipper name, broker name, start place, and delivery destination).
        ii. Once user added job, driver(“username” in display below) will receive job info from his/her mobile app.
    b. Edit Job
        i. User can edit job information such as shipper name, broker name, time, and addresses. However, when status of job is “PENDING” or “ACCEPTED”, user cannot reassign the job to another driver. If status is “DECLINED” user can reassign the job to another driver.
    c. Delete and Clear Job
        i. User can delete each job by selecting the certain job from list of jobs displayed in window.
        ii. Clear Job button will refresh the page.



id	username	shipper_name	broker_name	start_date	Origin	Destination	status
23	young	JJ	AA	05/05/19	Los Angeles	San Diego	ACCEPTED
20	young	Platinum	CalEarth	05/02/19	123	23	PENDING
22	young	So Cal Gradi...	CalEarth	04/13/19	Los Angeles	Las Vegas	ACCEPTED

Shipper Nam: JJ
Broker Name: AA
Username: jaejunmi
Date: 13/04/2019
Start Time: 12:00 AM
Rate: $ Per Hour
Origin Address:
Destination Address:
Comments:

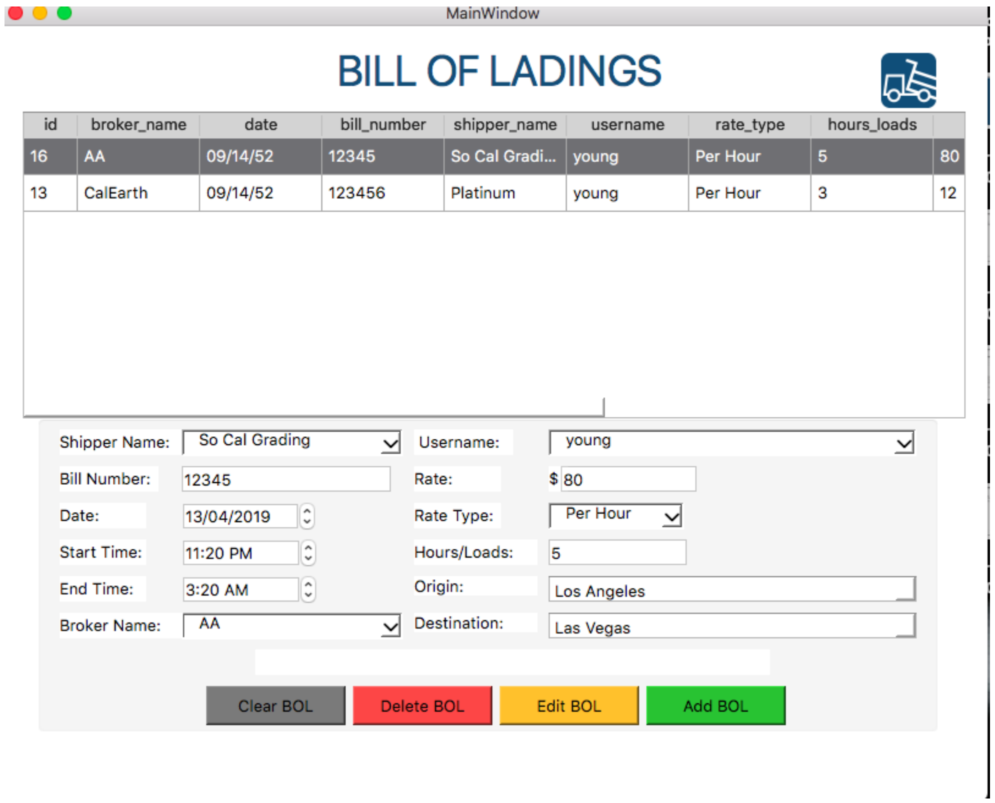
3. **Create Broker**
   a. User can add, delete, and edit broker information.
      i. User can edit and delete each broker's information(Name and Address) by selecting the certain broker from list of brokers shown on right side of window.
      ii. Clicking the "Clear Broker" button will refresh the page.
   b. Display window on right side will display the list of brokers.



3 - AA - fds
4 - CalEarth - 350 Commonwealth Ave Fullerton, CA 92832
2 - jjj - jjj
1 - ucla - fdsa

Broker Name: CalEarth
Broker Address: 350 Commonwealth Ave
Fullerton, CA 92832

4. **View Bill of Ladings**
    a. Display the list of bill of ladings.
    b. User can add, delete and delete the bill of ladings.
        i. User can manage completed job from this page.
            1. Invoice number(bill number), pay type, pay rate, and etc.
    c. Clear BOL button will refresh the page.



id	broker_name	date	bill_number	shipper_name	username	rate_type	hours_loads	
16	AA	09/14/52	12345	So Cal Gradi...	young	Per Hour	5	80
13	CalEarth	09/14/52	123456	Platinum	young	Per Hour	3	12

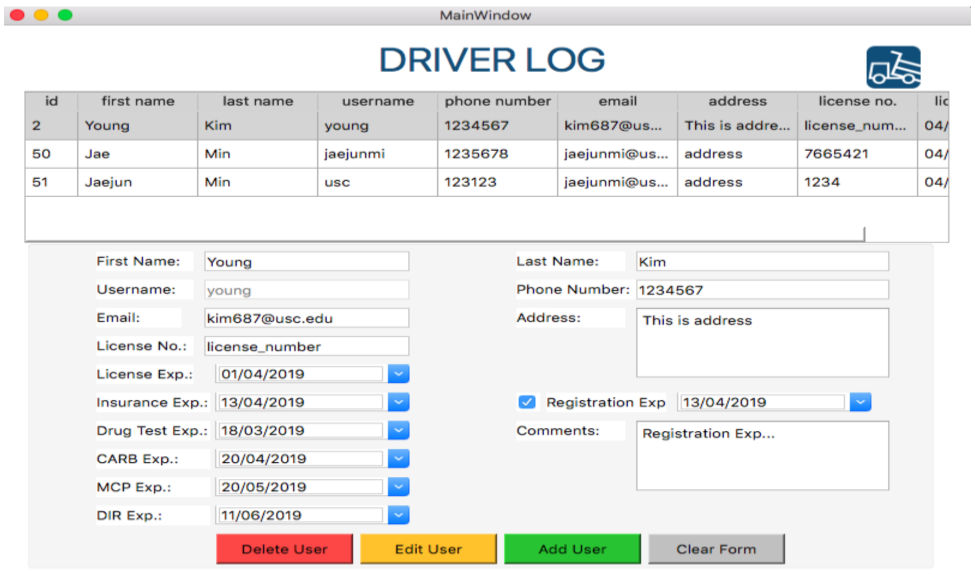
Shipper Name:	So Cal Grading	Username:	young
Bill Number:	12345	Rate:	$ 80
Date:	13/04/2019	Rate Type:	Per Hour
Start Time:	11:20 PM	Hours/Loads:	5
End Time:	3:20 AM	Origin:	Los Angeles
Broker Name:	AA	Destination:	Las Vegas

5. **Driver Info**
    a. User can add, delete and edit the driver.
        i. Add Driver:
            1. When user(stakeholder) add driver with username(driver's username), email, address and all the required licenses' expiration date, mobile app account for that driver is also created at same time with default password setting.
            2. After user(stakeholder) creates the account for the driver. Driver can change their account information from his/her mobile app.
        ii. Edit Driver:
            1. Once driver is added to driver log, user(stakeholder) can modify any information for the driver except for the username. User cannot change username.
        iii. Delete Driver:

1. User can delete the driver, and once driver is deleted by user, driver can no longer use that account.

iv. Clear From button will refresh the page.

v. Display window shows the list of drivers in table format.

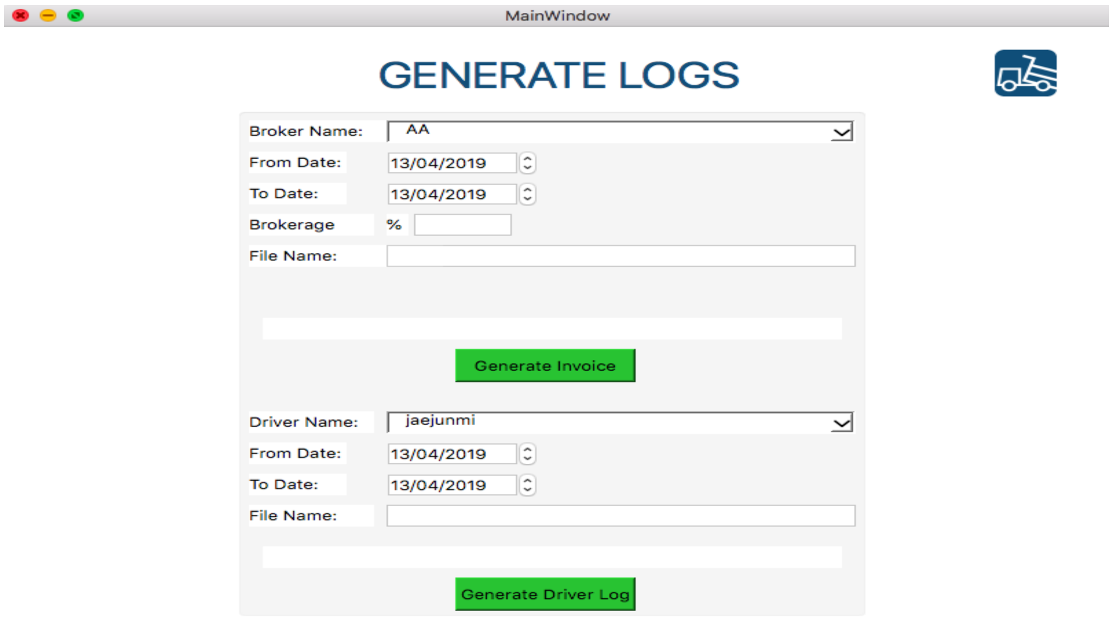


id	first name	last name	username	phone number	email	address	license no.	lic
2	Young	Kim	young	1234567	kim687@us...	This is addre...	license_num...	04/
50	Jae	Min	jaejunmi	1235678	jaejunmi@us...	address	7665421	04/
51	Jaejun	Min	usc	123123	jaejunmi@us...	address	1234	04/

First Name:	Young	Last Name:	Kim
Username:	young	Phone Number:	1234567
Email:	kim687@usc.edu	Address:	This is address
License No.:	license_number		
License Exp.:	01/04/2019		
Insurance Exp.:	13/04/2019	✓ Registration Exp	13/04/2019
Drug Test Exp.:	18/03/2019	Comments:	Registration Exp...
CARB Exp.:	20/04/2019		
MCP Exp.:	20/05/2019		
DIR Exp.:	11/06/2019		

6. Generate Logs

a. Generate Invoice creates the excel file with list of invoices base on user's selected option.

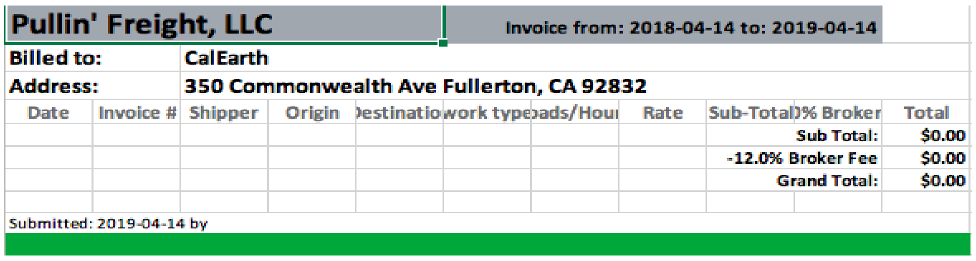


Broker Name:	AA
From Date:	13/04/2019
To Date:	13/04/2019
Brokerage	%
File Name:	

Driver Name:	jaejunmi
From Date:	13/04/2019
To Date:	13/04/2019
File Name:	

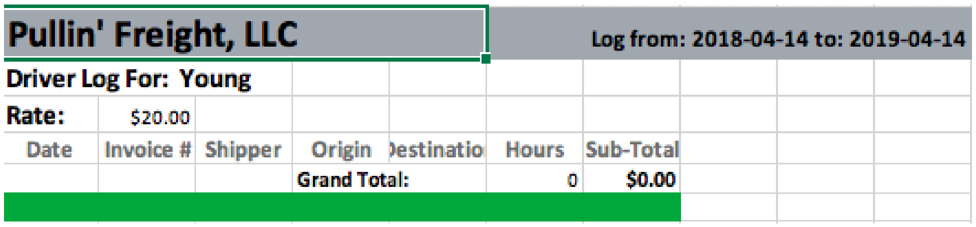
        i. User can generate invoice list base on broker name, dates, and brokerage.
    b. Generate Driver Log creates the excel file with list of jobs done by chosen driver.
        i. User can select the certain driver to extract their work information base on date.

Figure 1) Below image shows excel format of generate invoice.



Pullin' Freight, LLC					Invoice from: 2018-04-14 to: 2019-04-14					
Billed to:		CalEarth								
Address:		350 Commonwealth Ave Fullerton, CA 92832								
Date	Invoice #	Shipper	Origin	)estinatio	work type	)ads/Hour	Rate	Sub-Total	)% Broker	Total
									Sub Total:	$0.00
									-12.0% Broker Fee	$0.00
									Grand Total:	$0.00
Submitted: 2019-04-14 by										

Figure 2) Below image shows excel format of generate driver log.

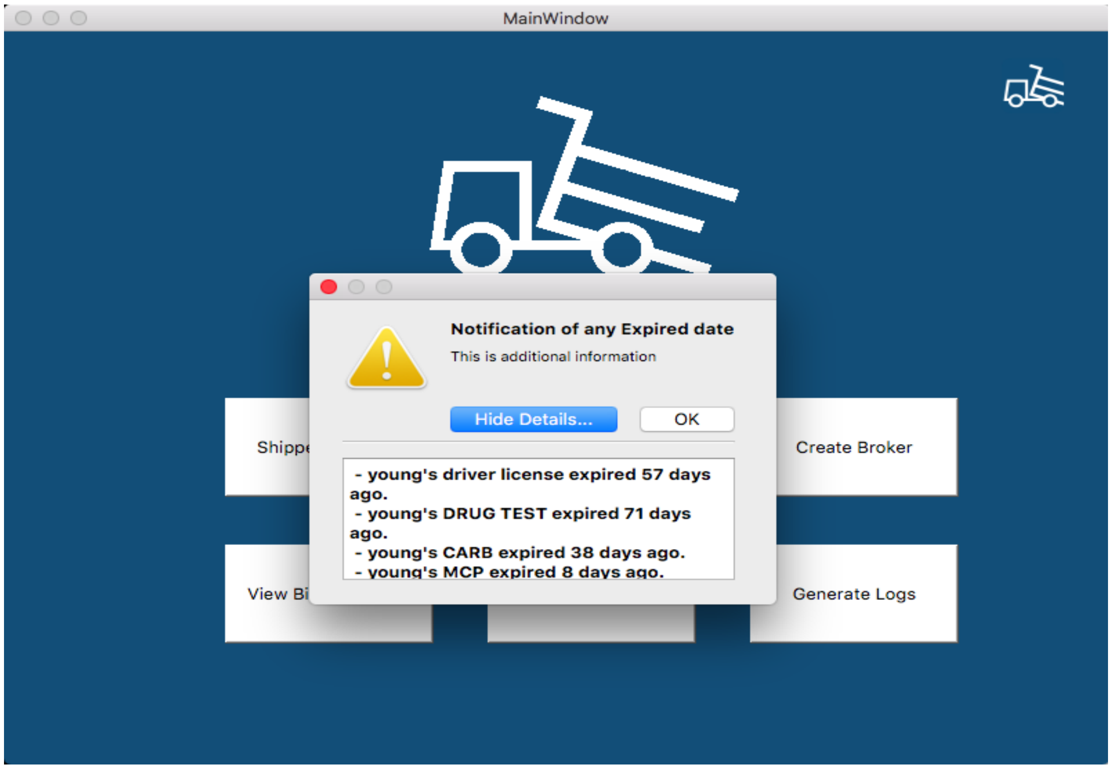


Pullin' Freight, LLC				Log from: 2018-04-14 to: 2019-04-14		
Driver Log For: Young						
Rate:	$20.00					
Date	Invoice #	Shipper	Origin	)estinatio	Hours	Sub-Total
			Grand Total:		0	$0.00

7. Notification

a. Notification window will popup by clicking the truck icon on upper right corner from the main page.

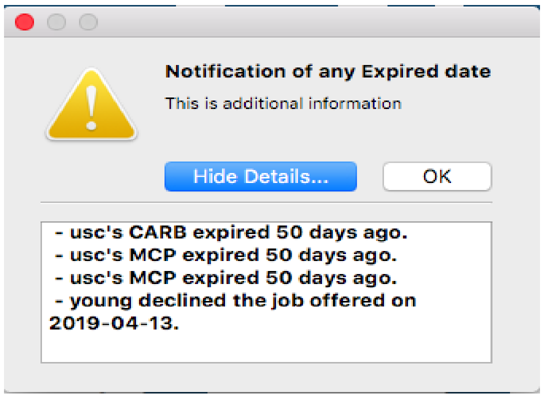
b. User is notified when drivers' license will be expire within 45 days or is already expired and when driver declined the job.



Notification of any Expired date
This is additional information

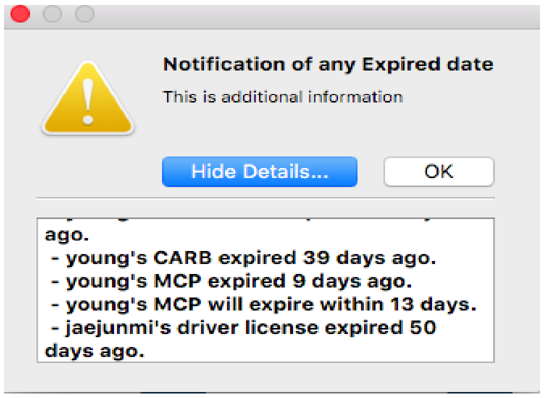
```
 - young's driver license expired 57 days
ago.
 - young's DRUG TEST expired 71 days
ago.
 - young's CARB expired 38 days ago.
 - young's MCP expired 8 days ago.
```

Figure) Left image in below shows when driver declined the job, and right image shows when driver's license is already expired or soon to be expired.



Notification of any Expired date
This is additional information

```
 - usc's CARB expired 50 days ago.
 - usc's MCP expired 50 days ago.
 - usc's MCP expired 50 days ago.
 - young declined the job offered on
2019-04-13.
```

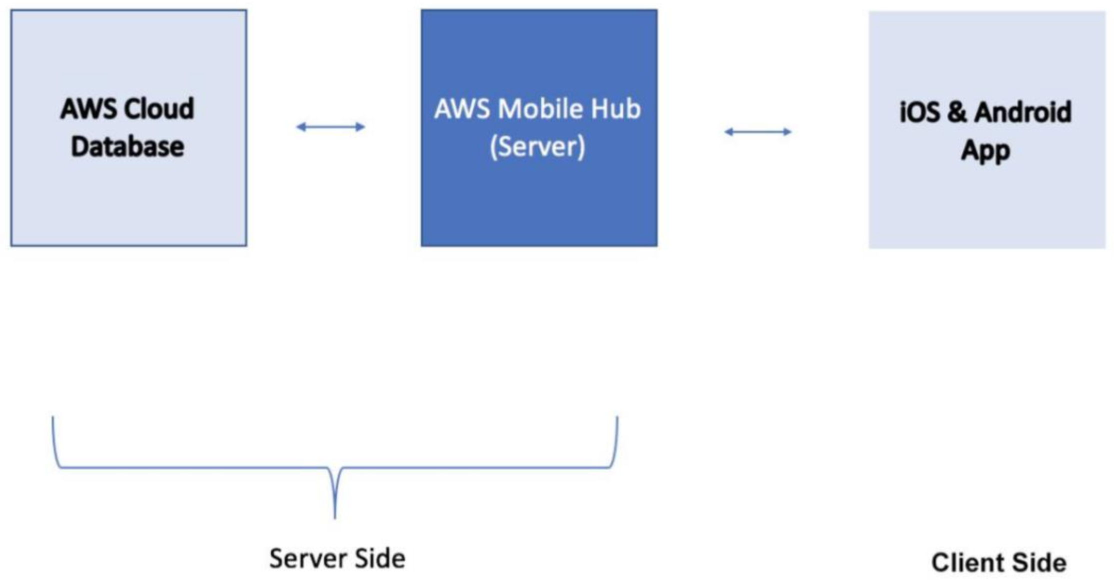


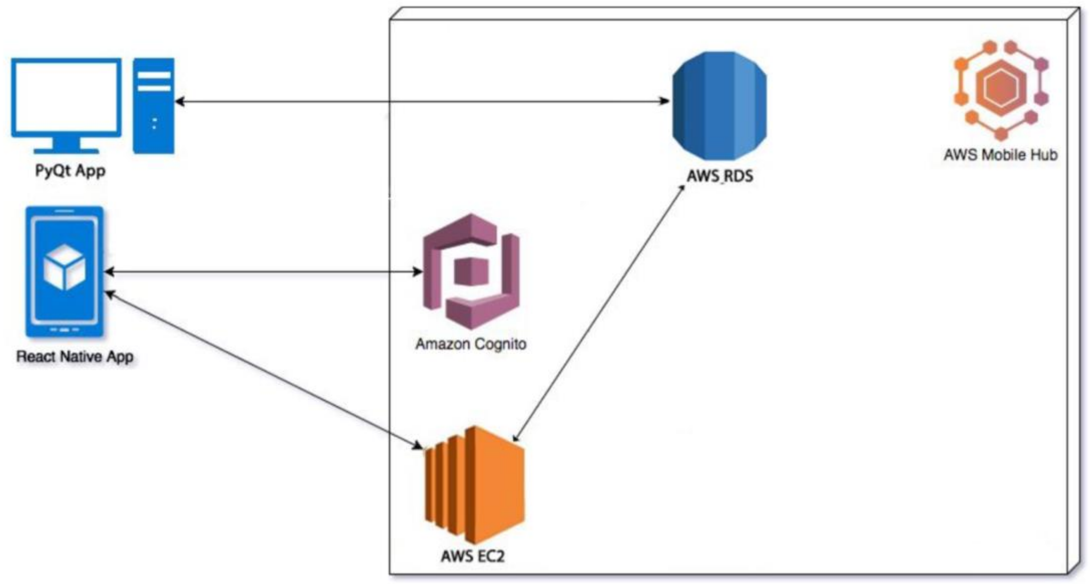
Notification of any Expired date
This is additional information

```
ago.
 - young's CARB expired 39 days ago.
 - young's MCP expired 9 days ago.
 - young's MCP will expire within 13 days.
 - jaejunmi's driver license expired 50
days ago.
```

System Architecture

In terms of system architecture, we followed the previous group's system architecture since this project was continuation from their project. Their principal design pattern is called "Three-Tier Pattern" which adopts "separations of concerns" design principle that splits the application into sections based on their functionalities. There are three components: Display tier, Business Logic tier, and Database tier. Display tier means all the visualization of application as well as the logic to update these screens. Business tier is mid-way server that handles the data retrieval and data update to the database. Lastly, database tier is MySQL database hosted by AWS RDS Database.





Server

Host: AWS EC2 Web Server
Scripting Language: PHP
Communication Mechanism: JSON(using 'fetch" fucntion)
Scripts:

1. DBConfig.php - This script configures with AWS database. Every scripts are using this script to use AWS RDS database.
2. getAccount.php - This script gets account information once user logs in.
3. getExpiration.php - This script gets all the expirations information from users table and expirations table.
4. getTodayJobs.php - This script gets the jobs that has its date attribution equal to current date.
5. getFutureJobs.php - This script gets all future jobs that were assigned to this user from the database.
6. getHistory.php - This script gets all of bill of ladings for this user from the database.
7. submitAccount.php - This script gets called when user press account change buttons. It will update the users database.
8. submitBill.php - This script adds bill of lading that was filled out by the user into the database.

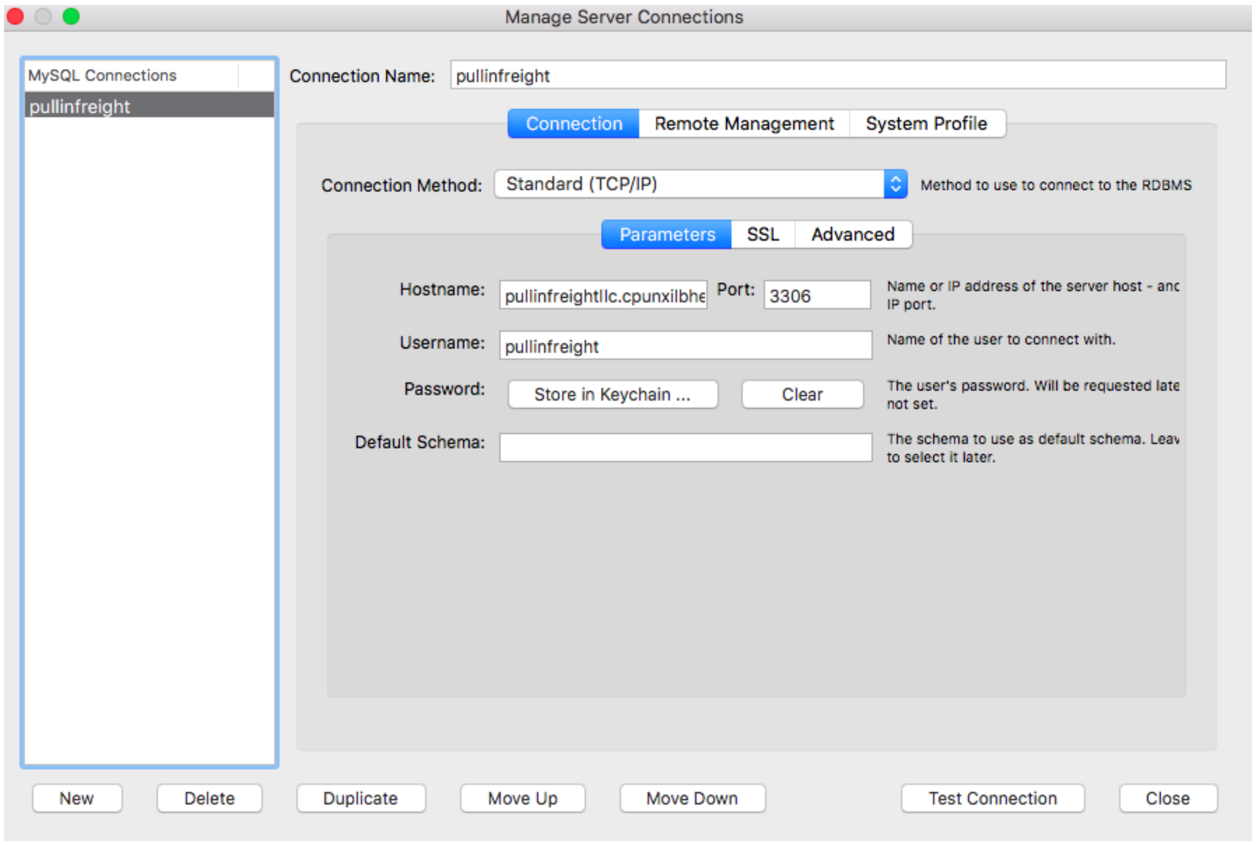
*Referenced from previous group

Database(AWS RDS - MySQL database)

Storage Allowed: 100.0 TB

Connection:

Hostname: pullinfreightllc.cpunxilbhe7v.us-west-1.rds.amazonaws.com
Connection: method: Standard (TCP/IP)
Username: pullinfreight
Port: 3306
*Password information stored in database_connector.py



pullinfreight

Connection Name: pullinfreight

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: pullinfreightllc.cpunxilbhe Port: 3306 — Name or IP address of the server host - anc IP port.

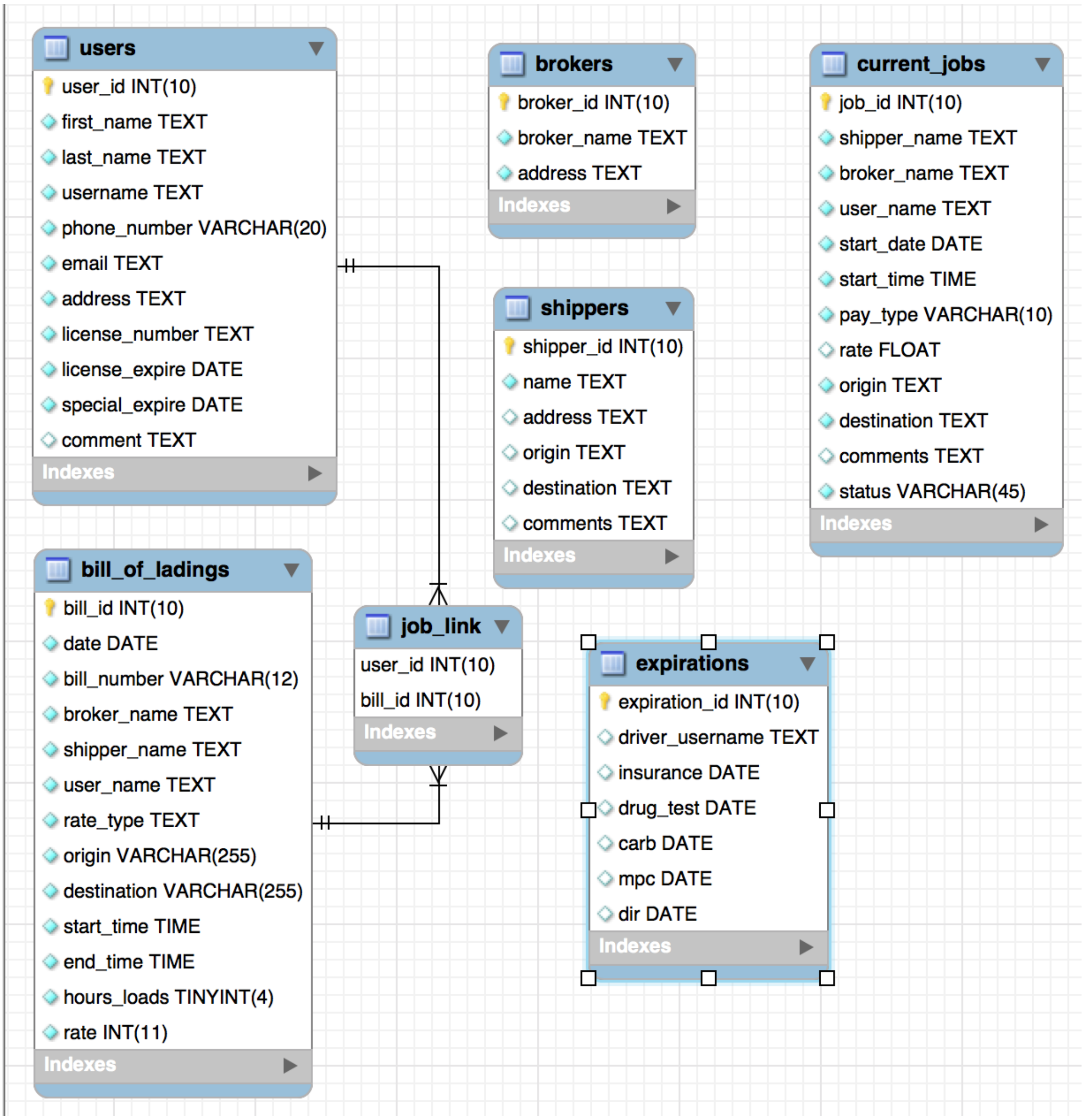
Username: pullinfreight — Name of the user to connect with.

Password: — The user's password. Will be requested late not set.

Default Schema: — The schema to use as default schema. Leav to select it later.

Tables

- Users - user data
- Brokers - brokers data
- Shippers - shippers data
- Bill_of_ladings - bill of ladings data
- Job_link - links bill of lading and user by their id
- Current_jobs - jobs data
- Expirations - expirations data for each user



Deployment

Mobile App:

The mobile application is developed using React-Native, javascript framework. Since this code can be converted into iOS as well as android, the application contains both packages. To run this application, use iOS or android mobile devices or application simulator using Xcode or Android Studio(device emulator must be installed on the machine). Node.js is also required for the simulation requirement. In order to install Node.js and npm which installs all of libraries required to run this application, type “**brew install node**” or “**sudo brew install node**” in the command line. This mobile application is not finished and need to be cleaned up in

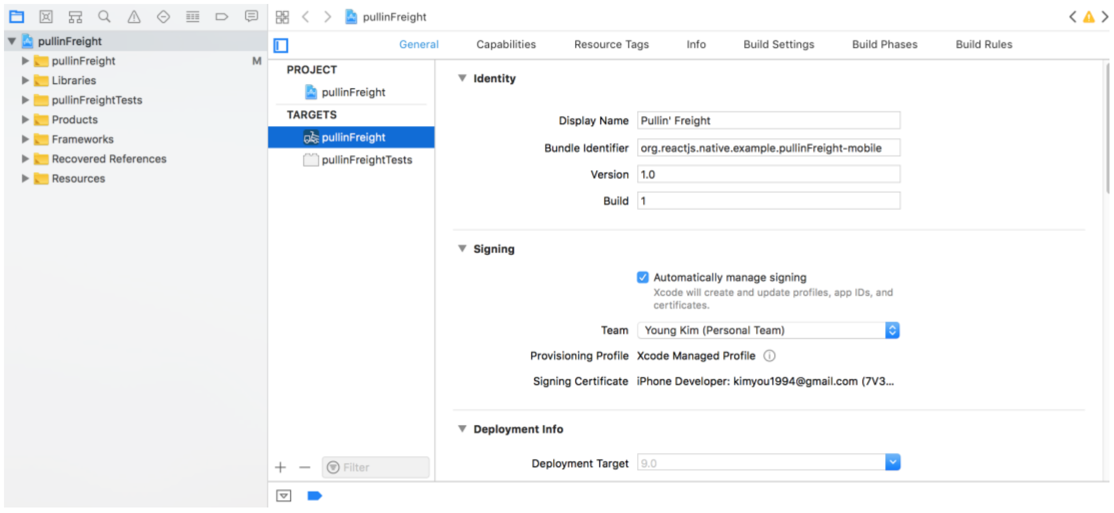
Setup Instruction for MacOS

To run on command line:

- Decompress mobile application zip file. Make sure you have all the contents. This application is using AWS-cognito so it required aws-export.js in the folder.
- In the command line, type “**npm install -g react-native-cli**” or “**sudo npm install -g react-native-cli**”.
- Navigate to the file location inside command line and type “react-native run-ios” or “react-native run-android” respectively for the desired emulation environment.

To run on phone device through Xcode:

- Navigate to ios folder and open Xcode named “**pullinFreight.xcodeproj**”.
- Connect iOS mobile device. In order to debug using mobile device, Signing need to be done using Apple ID. Shown by screenshot below, check “**Automatically manage signing**” and choose Team. Make sure the Bundle Identifier have a unique name.
- Press Build button.



Identity

Display Name: Pullin' Freight
Bundle Identifier: org.reactjs.native.example.pullinFreight-mobile
Version: 1.0
Build: 1

Signing

✓ Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team: Young Kim (Personal Team)
Provisioning Profile: Xcode Managed Profile
Signing Certificate: iPhone Developer: kimyou1994@gmail.com (7V3...

Deployment Info

Deployment Target: 9.0

Some helpful resources for React Native

Other means of running the application in a simulated environment can be found on:

https://facebook.github.io/react-native/docs/getting-started.html

For simulation in Windows and Linux environment and further details please refer to:
https://facebook.github.io/react-native/docs/getting-started.html

For further troubleshooting informations, please refer to:
https://facebook.github.io/react-native/docs/troubleshooting#content

Standalone App:

The Operation Dashboard Pullin’ Freight is desktop app developed using Python3 & PyQt5. The executable can only been run in Unix environments (Not working on Windows).

Setup instructions for MacOs:

To compile the Operation Dashboard:

- Install python3, In command line type, “**brew install python3**”
- Once python 3 is installed, type “python3 <filename>” in command line from current program directory to compile the program. In our case: “**python3 home.py**”.

To make an executable file for user:

- After development is completed, use pyinstaller to create the executable file.
    - In command line type, “**pip install pyinstaller**”
- In the command line type “**pyinstaller --onefile home.py**” to create executable file. Once file is created, file should be located inside “dist” folder.
- After create the executable file “home.spec” will be created in directory where “home.py” is located. Modify “home.spec” to properly load the images in executable application.
    - For example add: a.datas += [('truck_blue.png','./image/truck_blue.png', "DATA")]
- Once you modified the .spec file, in command line, type “ pyinstaller --onefile --clean home.spec”
- Go to “dist folder” and double click the “home” executable to start the program.

Some helpful resources for pyinstaller

https://pyinstaller.readthedocs.io/en/v3.3.1/usage.html