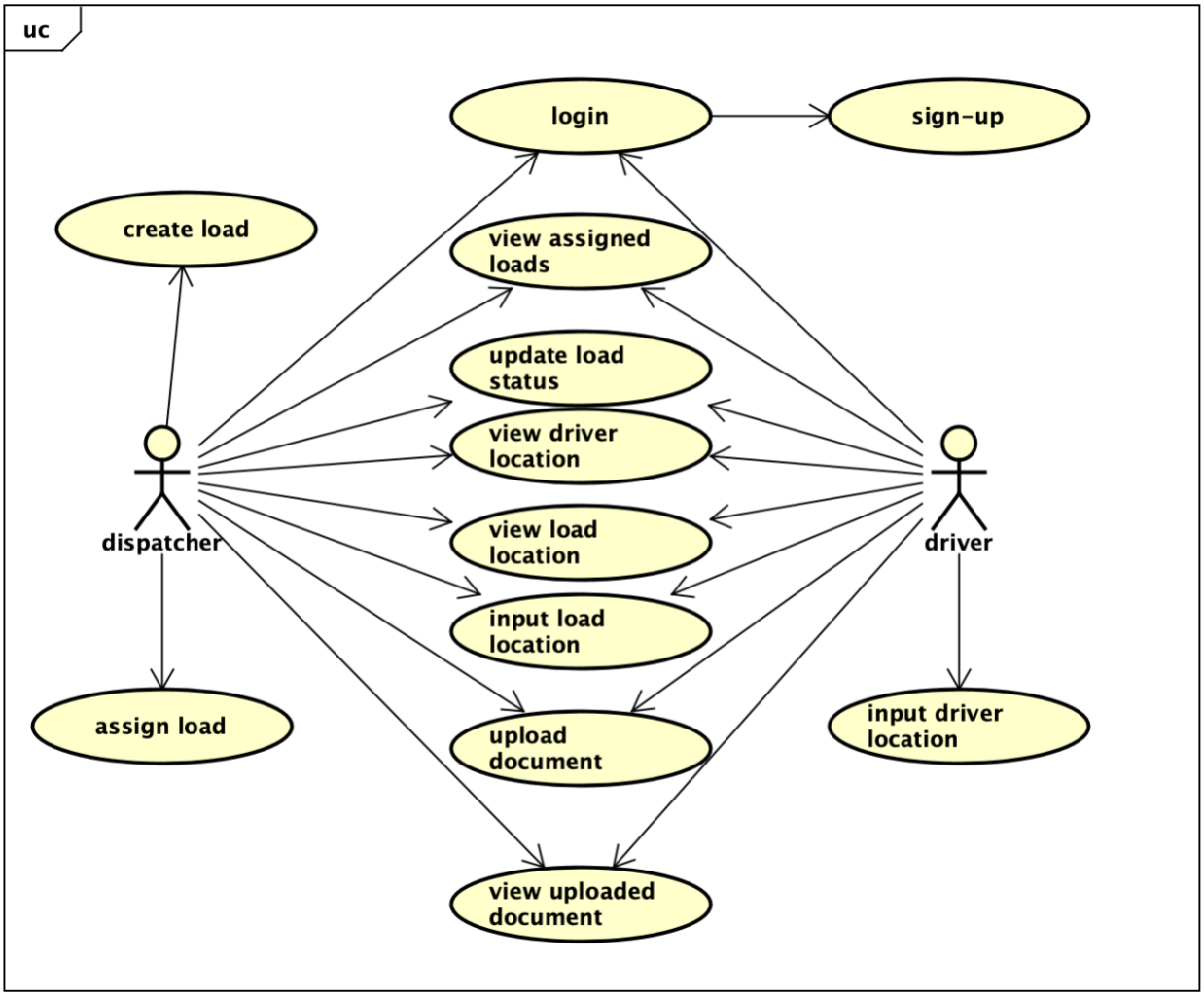
CSE 335 Class Project Phase I
Submission Template

Team Member Names: Shubhdeep Kaur

a) Copy and paste your Use case diagram here



b) Description of each major use case here

ITEM	VALUE
UseCase	create load
Summary	Dispatcher creates a new load entry with pickup, drop-off, and cargo details .
Actor	dispatcher
Precondition	Dispatcher must be logged in.
Postcondition	New load is created and stored in Firebase.
Base Sequence	1. Dispatcher presses create load button 2. enters pickup, drop-off, cargo details, and estimated delivery date 3. press done 4. FIrebase saves detai ls
Branch Sequence	press cancel button and return to dashboard
Exception Sequence	missing input field leads to message
Sub UseCase	
Note	

ITEM	VALUE
UseCase	assign load
Summary	Dispatcher assigns an existing load to a specific driver.
Actor	dispatcher
Precondition	Dispatcher is logged in and there exists an unassigned load and one driver
Postcondition	Load is assigned and visible in driver's dashboard.
Base Sequence	1. Dispatcher selects unassigned load from list view 2. chooses a driver fro m list 3. confirms assignment 4. system updates
Branch Sequence	Dispatcher searches drivers by name or availability.
Exception Sequence	none
Sub UseCase	
Note	Firebase gets updated, drivers dashboard must also update immediately

ITEM	VALUE
UseCase	login
Summary	Allow dispatchers and drivers to login using Firebase to get access to dash board based on their role.
Actor	dispatcher, driver
Precondition	User account exists.
Postcondition	User is successfully logged in and taken to either driver or dispatcher dashb oard.
Base Sequence	1. User opens app and sees login prompts 2. User enters details and press es login button 3. System uses Firebase to authenticate 4. System fetches user role 5. User taken to appropriate dashabord
Branch Sequence	User enters sign up and uses sign up use case
Exception Sequence	wrong login leads to error message
Sub UseCase	
Note	Created load remains unassigned until dispatcher selects a driver.

ITEM	VALUE
UseCase	view assigned loads
Summary	Both dispatchers and drivers can view assigned loads in list form.
Actor	dispatcher, driver
Precondition	user is logged in and load exists in Firebase
Postcondition	List of relevant loads displayed in TableView.
Base Sequence	1. user opens my loads list 2. fetch data from Firebase 3. list is displayed with details
Branch Sequence	dispatcher filters by driver name, driver filters my status
Exception Sequence	no loads found leads to no active loads
Sub UseCase	
Note	uses Firebase and refresh instantly

ITEM	VALUE
UseCase	update load status
Summary	Allows dispatcher or driver to change a load's current progress status.
Actor	dispatcher, driver
Precondition	load is assigned and user is logged in
Postcondition	Status updated in Firebase and reflected in dashboards.
Base Sequence	1. user selects load 2. choose status such as on the way, waiting, loading, unloading, completed 3. system updates
Branch Sequence	Dispatcher updates status manually
Exception Sequence	none
Sub UseCase	
Note	Firebase updates

ITEM	VALUE
UseCase	view driver location
Summary	Displays the real-time or last known location of the driver on a map.
Actor	dispatcher, driver
Precondition	Driver has inputted or shared their location
Postcondition	Map updates to show driver's current or recent position
Base Sequence	1. user selects view map button 2. system gets drivers location using Fireb ase 3. map shows location with marker
Branch Sequence	dispatcher selects different driver to view
Exception Sequence	location not available if driver hasn't shared
Sub UseCase	
Note	Map implemented with MapKit API using stored coordinates.

ITEM	VALUE
UseCase	view load location
Summary	Shows pickup and drop-off locations of a specific load on a map.
Actor	dispatcher, driver
Precondition	load location has already been inputted
Postcondition	user sees pickup and dropoff points on map
Base Sequence	1. user opens load on load detail view 2. app fetches location data 3. map shows locations
Branch Sequence	Driver chooses navigation to destination.
Exception Sequence	no location inputted leads to no location
Sub UseCase	
Note	Uses Apple MapKit API for displaying and routing between load points.

ITEM	VALUE
UseCase	input load location
Summary	Allows dispatcher or driver to input or edit pickup/drop-off coordinates.
Actor	dispatcher, driver
Precondition	user logged in
Postcondition	new coordinated saved in Firebase
Base Sequence	1. user selects load and chooses edit location button 2. enters location usin g words or select 3. confirm button 4. system save to Firebase
Branch Sequence	none
Exception Sequence	location not found warning
Sub UseCase	
Note	Load location data shared across both dispatcher and driver dashboards.

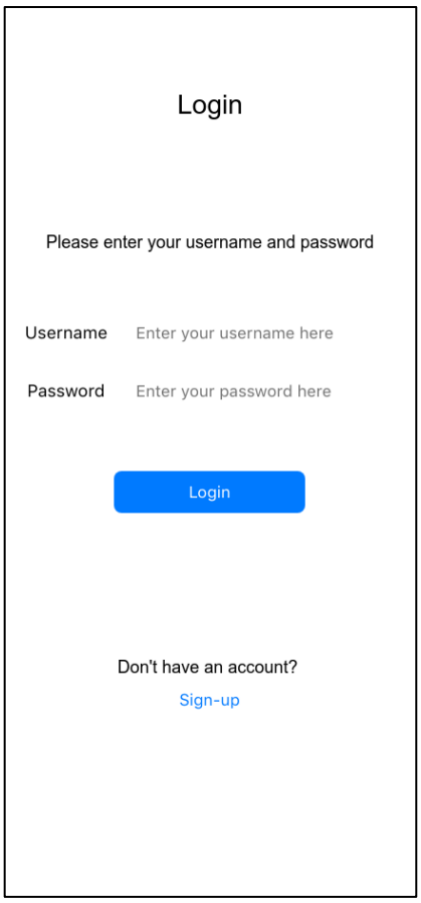
ITEM	VALUE
UseCase	upload document
Summary	User uploads delivery-related documents (such as bill of lading, receipts)
Actor	dispatcher, driver
Precondition	user logged in and load selected
Postcondition	document uploaded to Firebase and linked to driver ID
Base Sequence	1. user opens documents 2. selects image/pdf to upload 3. system update to Firebase 4. upload displayed
Branch Sequence	cancel upload and go back to dashboard
Exception Sequence	invalid upload
Sub UseCase	
Note	Dispatcher and driver can both access uploaded documents.

ITEM	VALUE
UseCase	view uploaded document
Summary	Displays uploaded documents related to a specific load
Actor	dispatcher, driver
Precondition	at least one document uploaded for that specific load
Postcondition	document displayed
Base Sequence	1. user opens documents view 2. file is fetched 3. display the document 4. user can also add any more documents
Branch Sequence	can display by uploaded date
Exception Sequence	no current documents message
Sub UseCase	
Note	documents saved in Firebase Storage

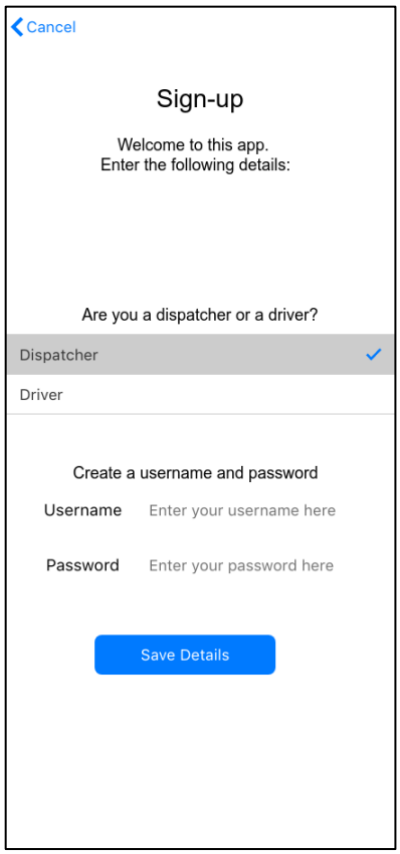
ITEM	VALUE
UseCase	sign-up
Summary	Allows a new user to create an account using user and password
Actor	
Precondition	user cannot login because doesn't have account yet
Postcondition	new ser account registered in sytem and with either dispatcher or driver role
Base Sequence	1. go to sign up page from login page 2. enter details such as user, password, and role 3. Firebase creates new account for user 4. confirm button and saved
Branch Sequence	if user exists then display message, promt user to go back to login page
Exception Sequence	none
Sub UseCase	
Note	All registered users must have a valid username and password to access system functions.

ITEM	VALUE
UseCase	input driver location
Summary	Driver inputs their location manually
Actor	driver
Precondition	driver is logged in and location on
Postcondition	coordinates updated and saved to firebase
Base Sequence	1. driver has location on 2. driver inputs location 3. saved to firebase
Branch Sequence	submit/share button
Exception Sequence	no location found
Sub UseCase	
Note	Driver manually updates location

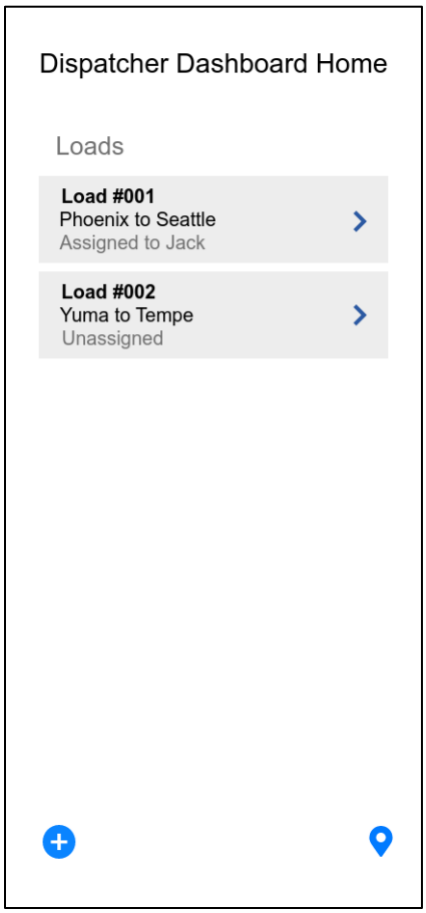
c) UI Walkthrough: Copy and paste each UI screen mock and add a description of each UI mockup screen.



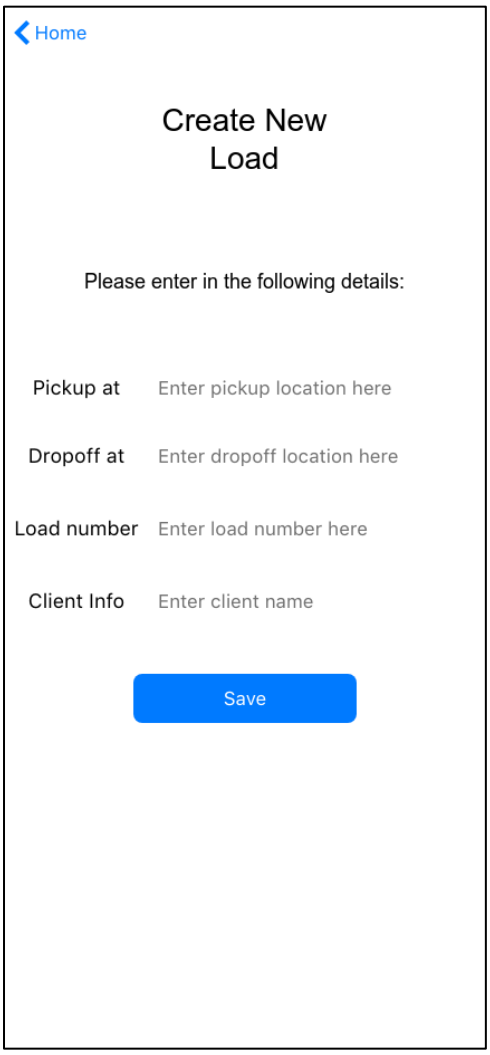
This is the login screen and the first screen the user sees. The user can either type their username and password in if they already have an account or press the sign-up button.



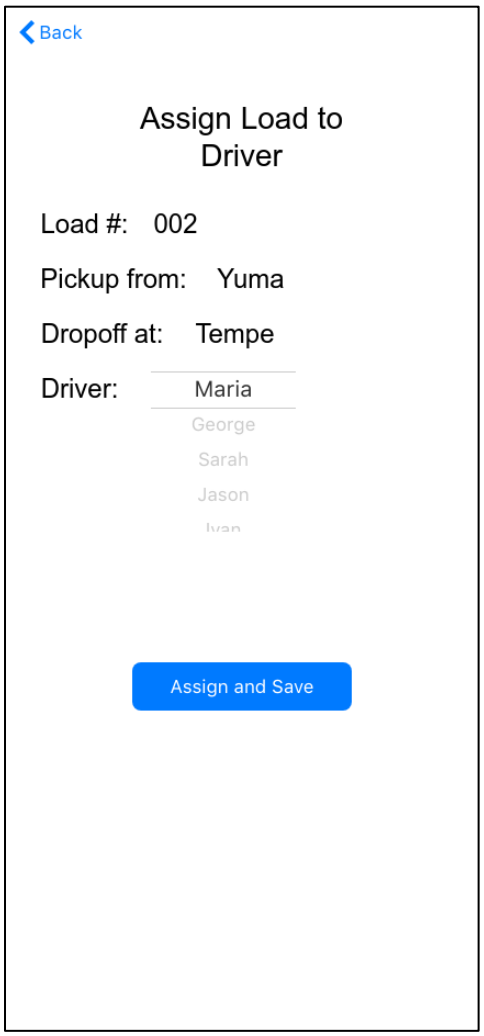
This is the sign-up screen where the user who isn't registered into the system creates their account. They are prompted to select whether they are a dispatcher or driver, and then also input their username and password that they will use. They can cancel this task and go back to login screen or they can press the save button and go back to the login screen to login.



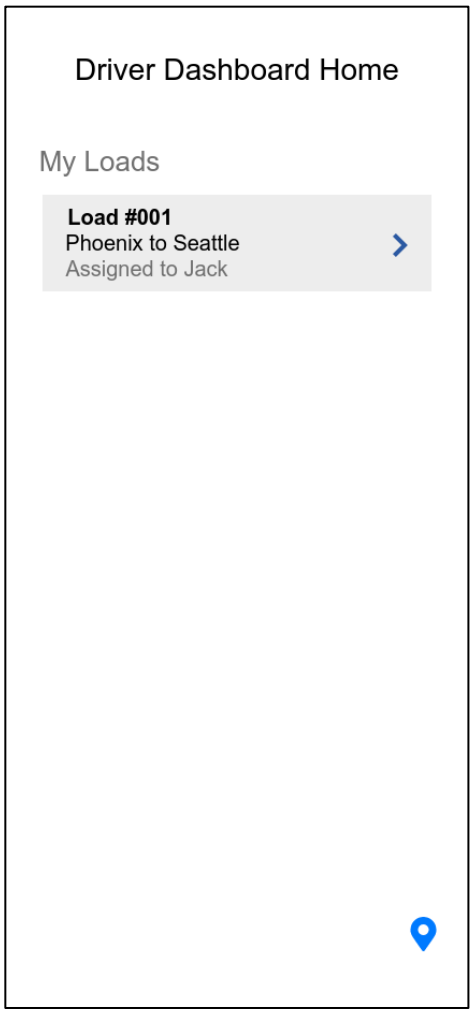
If the user account is a dispatcher, after each login they will be directed to this main dashboard homepage. This dashboard consists of a list of loads within their company with key details like load number, pickup to dropoff location, and the driver it's assigned to. The dispatcher can create new loads, see the drivers' locations on the map, as well as get a detailed view of each load.



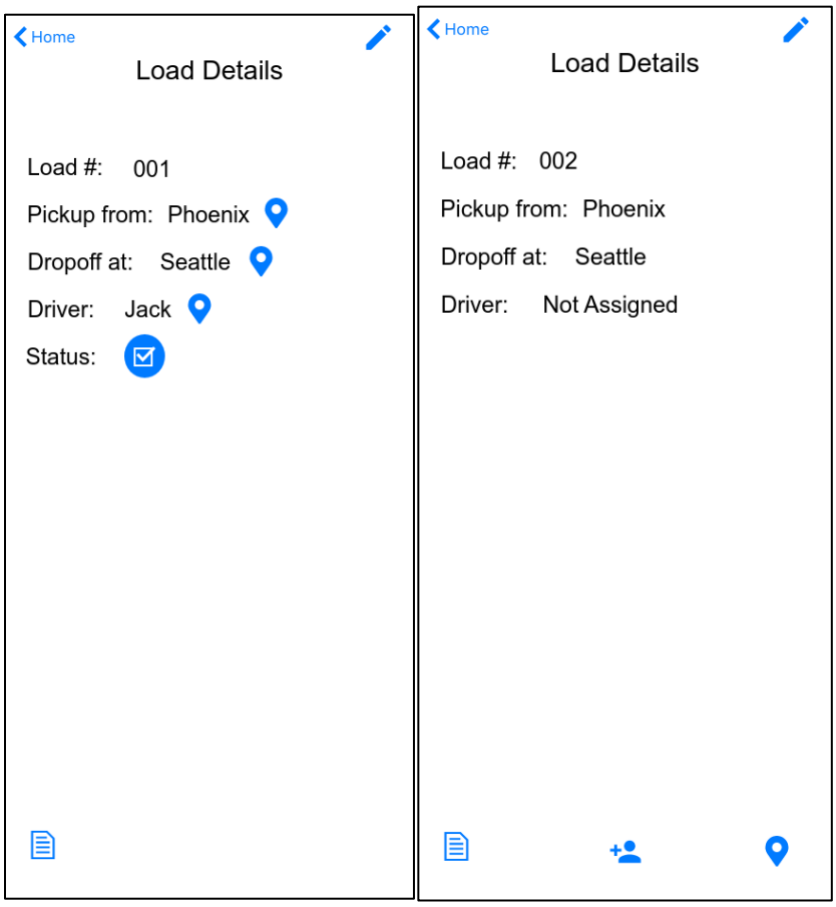
This create load screen is a task that the dispatcher can complete only. This screen allows the dispatcher to add a new load to the load list and input in details such as pickup location, dropoff location, load number, and client info. They can go back to home page if they change their mind, or they can press save button once they're done.



This assign load screen is used by dispatchers to assign an available load to a driver. The screen has the load details loaded up and requires the dispatcher to select a driver from their company to be assigned to the load. Once chosen, the user can press the save button.

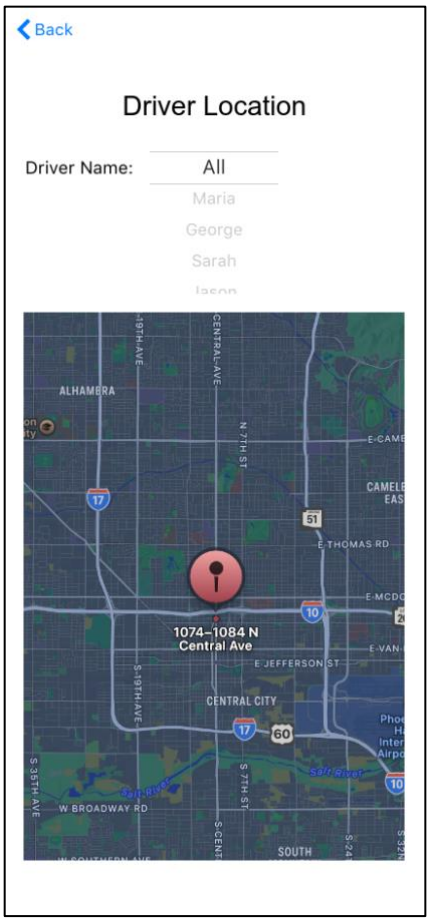


This is the screen that serves as a homepage for a driver user account. The driver can see their load that is assigned to them and its details and edit the status or view the locations as well as go to a map view of their own location.

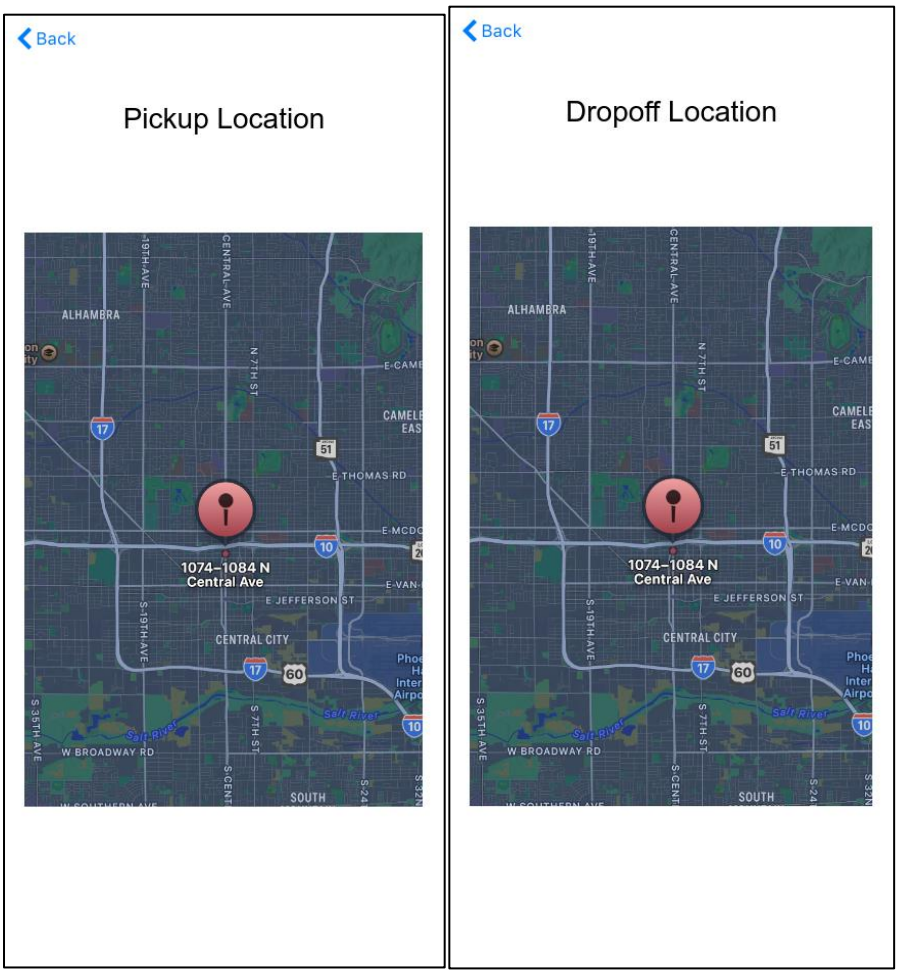


This load details screen is what shows up when any user goes into the load details. It shows the load info such as load number, pickup location, dropoff location, driver name, and status, as well as an option on the bottom to see documents. The user can also see the location of the pickup location as well as the dropoff location on a map. The driver's location can also be seen. The driver or the dispatcher can also update the status of the load by checking or unchecking the checkmark.

The screen on the right shows a case when the load doesn't have an assigned driver yet and the dispatcher can press the add person icon at the bottom to assign a driver to the load.

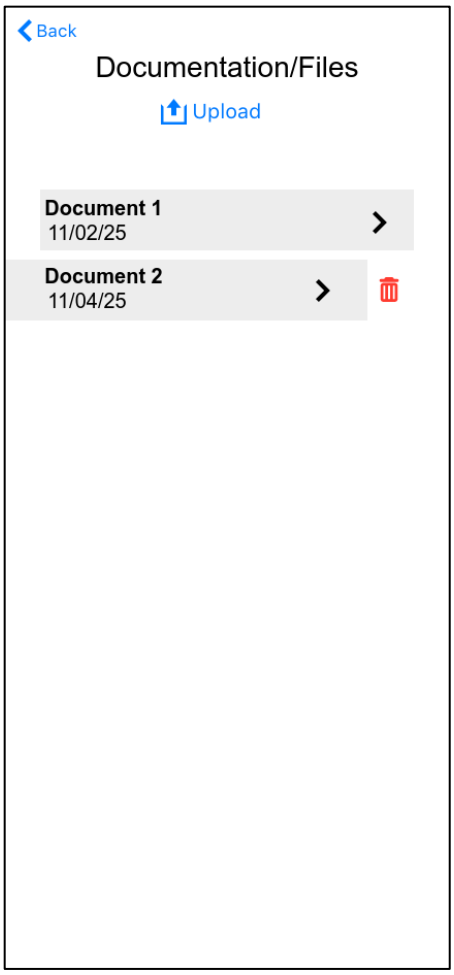


This screen is shown as a result of pressing the map icon next to the driver name, or pressing the map icon on the dispatchers' dashboard. The driver or the dispatcher can select a driver name to show their location.

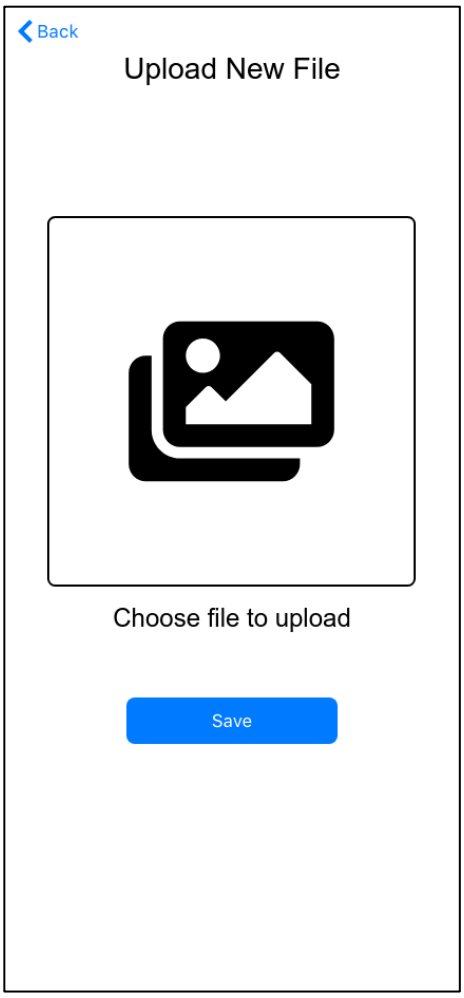


The screen on the left is shown when the map icon next to the pickup location is pressed from the load detail view. The user can see the location on apple map.

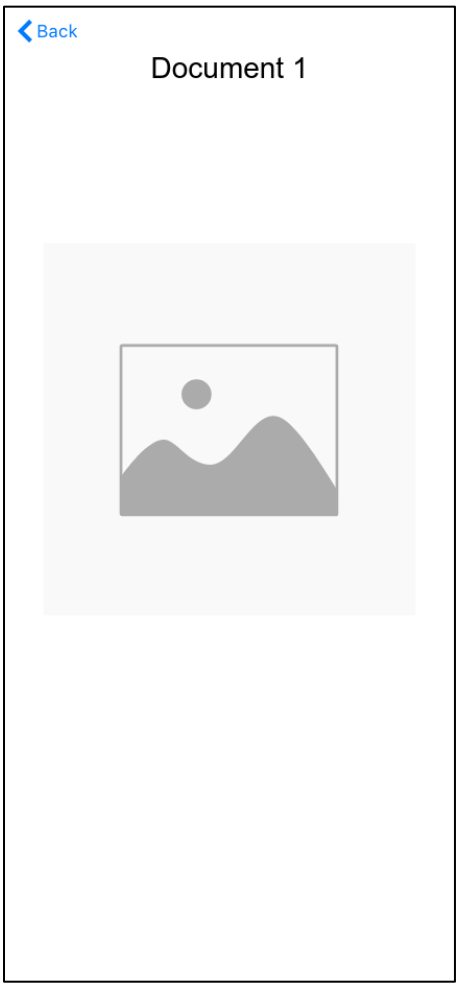
The screen on the right is shown when the map icon next to the dropoff location is pressed from the load detail view. The user can see the location on apple map.



This screen is shown when the user was in load detail view and presses the document icon. The dispatcher or driver can see a list of documents pertaining to a particular load in a list style. There is an option to upload new files as well as to view the document. The document list is sorted based on date from oldest to newest. If the user swipes to the left, it will delete the document. Preferably, a confirmation will show up before it is permanently deleted from the system.

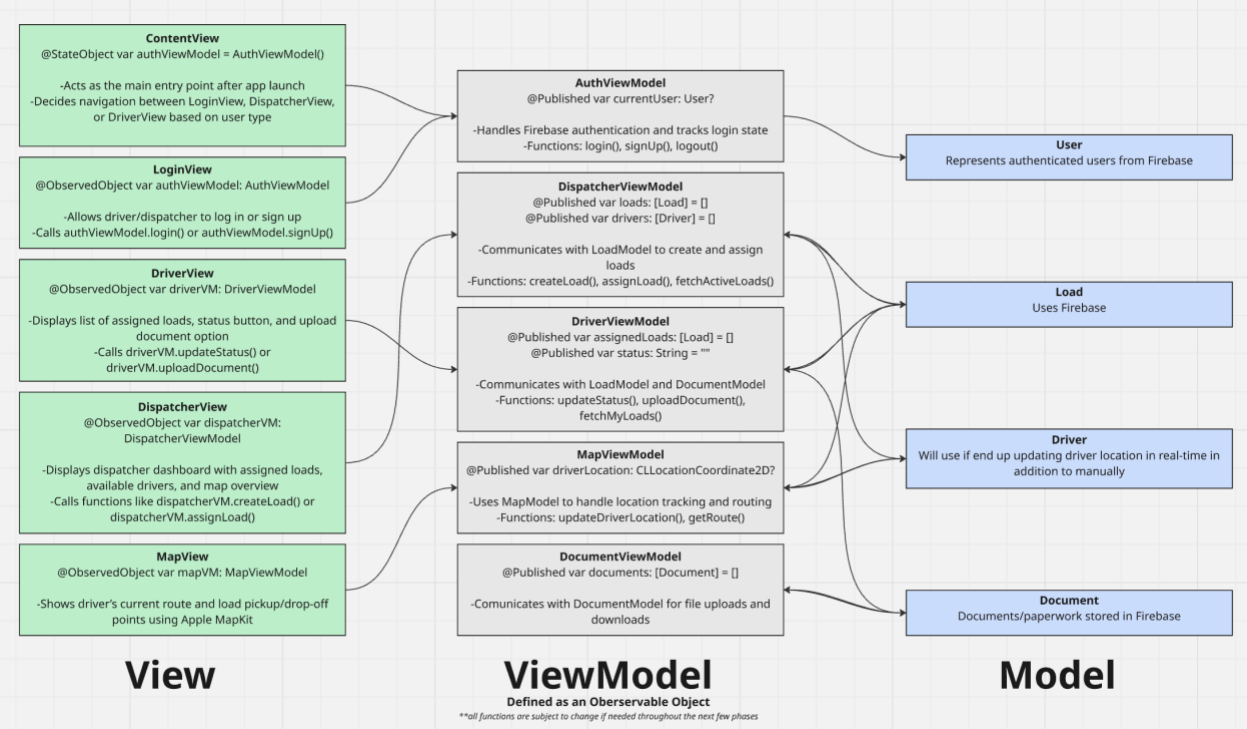


This is the upload new file/document screen where both drivers and dispatchers can upload any important or relevant documents related to a load. The user presses the big photo icon and chooses what they want to upload in from their files. The user then presses the save button to ensure that the file is saved into the system.



This screen is the document view screen which shows up as a result of the user pressing the chevron right icon next to the document from the document list screen. The user simply sees the name of the document and a big view of the document on their screen. The user can go back using the icons on the top left.

d) Copy and Paste your Initial MVVM Diagram here



e) Upload your 1st iteration of the implementation into the google drive and provide a link for the google drive.
https://drive.google.com/drive/folders/1WQbP-7pgcZuKkhh-fh12LLola-jH96_W?usp=sharing

f) Record the demo of your application after the implementation of the 1st iteration and provide a link for the demo (you can upload your recording to the YouTube and provide the YouTube link)
https://youtu.be/OHkhAtyjMWk