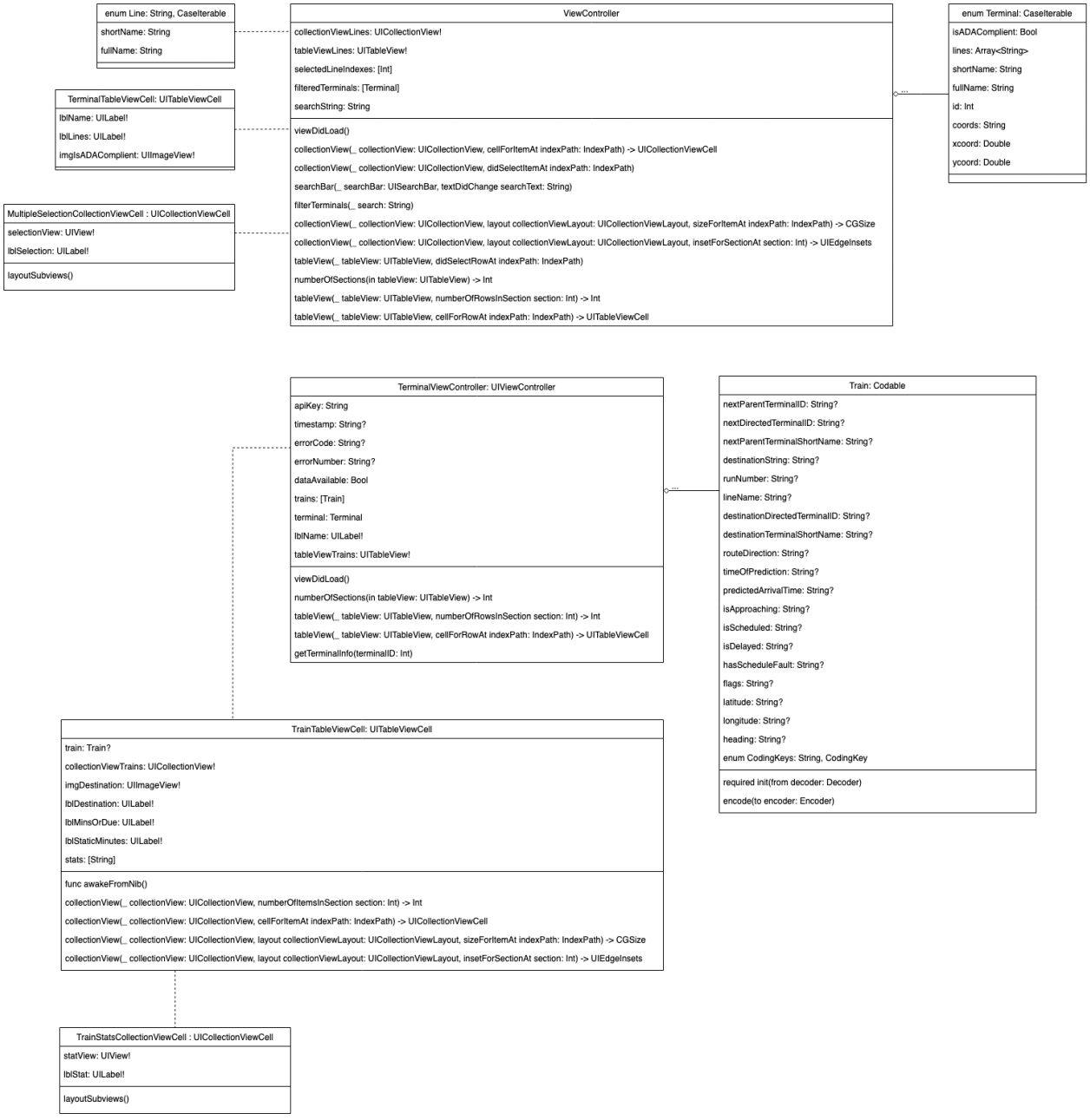
CSC491 Blake Boris

As I began assignment 2, I had just chosen my final project idea, which also involves parsing JSON data from the CTA. Therefore, I wanted to go all out on assignment 2 to parse the data correctly the first time. I started by making some basic calls to the train tracker API from my browser and documenting all the eta response items for myself in a custom swift class, *Train*, that implements the Codable protocol. Once I was able to properly serialize the incoming data for one station (or *terminal*), I knew I needed a comprehensive way to represent every L station within the L system. Luckily, the CTA's documentation for their APIs includes a CSV file containing all of the L stops along with their respective GHFS codes and location data. The spreadsheet also includes a table of true false values for which lines correspond with which stops. I wrote a quick script in Python to reorganize all of the data into Tuples and to output lines upon lines of Swift code to create another custom class, *Terminal*, which houses all the information contained in that spreadsheet. Not only that, but the reorganization of the data in Python allowed me to combine some of the more granular data about stations and directly associate each station with the lines it services by converting the table of true false values to a list of line names.

My next focus was UI. Once again, I wanted to go all out to make my life easier when I get to work on my final project. The idea of selecting one or multiple lines seemed really appealing to me, but I had never implemented a multi-select element in Swift, and there didn't seem to be a good out-of-box solution in Xcode. Therefore, I implemented my own version of a horizontally scrolling UICollectionView. I had originally wanted it to have multiple lines of options to fit more lines on screen at once, but the spacing wasn't uniform across each line, so I eventually settled on a single row of lines, and I think it was for the best.

Finally, after the user selects one or more lines, and even after making a possible search for a specific station name, they can select a station. The app then makes a call to the CTA's train tracker API to get information on upcoming trains for that station, and each of the etas fetched from the API are listed out in a UITableView. Not only that, but I was able to nest another custom UICollectionView to cleanly display key information about each train alongside the estimated minutes until arrival based on the serialized JSON data. Overall, I'm extremely happy with how this assignment turned out, and my friends even said my UI looked professional. I hope to expand on the information I've already gathered both from the CSV and the JSON response data for my final project, and I'm really looking forward to better predicting arrival times and improving the experience of estimation for the end user in my final project.


enum Line: String, CaseIterable
shortName: String
fullName: String
TerminalTableViewCell: UITableViewCell
lblName: UILabel!
lblLines: UILabel!
imgIsADACompliant: UIImageView!
MultipleSelectionCollectionViewCell : UICollectionViewCell
selectionView: UIView!
lblSelection: UILabel!
layoutSubviews()
ViewController
collectionViewLines: UICollectionView!
tableViewLines: UITableView!
selectedLineIndexes: [Int]
filteredTerminals: [Terminal]
searchString: String
viewDidLoad()
collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell
collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath: IndexPath)
searchBar(_ searchBar: UISearchBar, textDidChange searchText: String)
filterTerminals(_ search: String)
collectionView(_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, sizeForItemAt indexPath: IndexPath) -> CGSize
collectionView(_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, insetForSectionAt section: Int) -> UIEdgeInsets
tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath)
numberOfSections(in tableView: UITableView) -> Int
tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int
tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell
enum Terminal: CaseIterable
isADAComplient: Bool
lines: Array<String>
shortName: String
fullName: String
id: Int
coords: String
xcoord: Double
ycoord: Double
TerminalViewController: UIViewController
apiKey: String
timestamp: String?
errorCode: String?
errorNumber: String?
dataAvailable: Bool
trains: [Train]
terminal: Terminal
lblName: UILabel!
tableViewTrains: UITableView!
viewDidLoad()
numberOfSections(in tableView: UITableView) -> Int
tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int
tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell
getTerminalInfo(terminalID: Int)
Train: Codable
nextParentTerminalID: String?
nextDirectedTerminalID: String?
nextParentTerminalShortName: String?
destinationString: String?
runNumber: String?
lineName: String?
destinationDirectedTerminalID: String?
destinationTerminalShortName: String?
routeDirection: String?
timeOfPrediction: String?
predictedArrivalTime: String?
isApproaching: String?
isScheduled: String?
isDelayed: String?
hasScheduleFault: String?
flags: String?
latitude: String?
longitude: String?
heading: String?
enum CodingKeys: String, CodingKey
required init(from decoder: Decoder)
encode(to encoder: Encoder)
TrainTableViewCell: UITableViewCell
train: Train?
collectionViewTrains: UICollectionView!
imgDestination: UIImageView!
lblDestination: UILabel!
lblMinsOrDue: UILabel!
lblStaticMinutes: UILabel!
stats: [String]
func awakeFromNib()
collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int
collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell
collectionView(_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, sizeForItemAt indexPath: IndexPath) -> CGSize
collectionView(_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, insetForSectionAt section: Int) -> UIEdgeInsets
TrainStatsCollectionViewCell : UICollectionViewCell
statView: UIView!
lblStat: UILabel!
layoutSubviews()