MIAMI

CSC 431

EDGAR Terminal

System Architecture Specification (SAS)

Team 20

Alexander Arevalo	<Role>
Daniel Benayoun	<Role>
Jonathan M Brooks	<Role>
<Member Name>	<Role>

Version History

Version	Date	Author(s)	Change Comments
1	4/1/21		First draft
2	4/13/21		Second Draft
3	4/30/21		Third Draft, fixed some problems stated in Slack

Table of Contents

Table of Tables

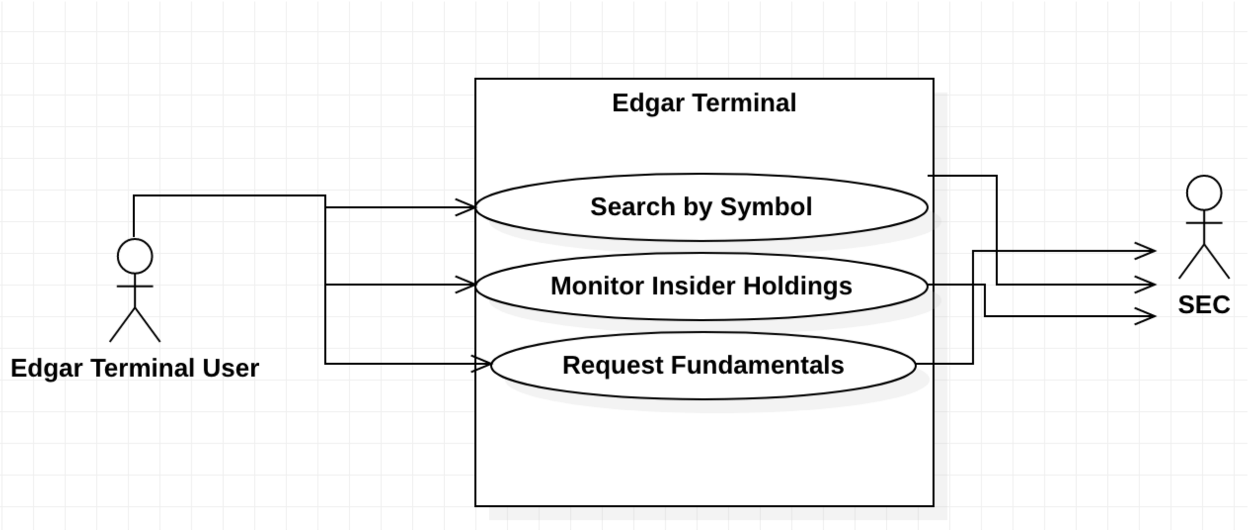
Table of Figures

111.System Analysis

111.1 System Overview

The system consists of three main parts: search for the stock by symbol; view insider holdings; and request fundamentals. In the system, the user will enter a stock trading symbol, the symbol is resolved into CIK number, subsequent files are pulled from the SEC database using a web scraper. The data received is then formatted to json using an XBRL parser. A user will be able to see our stock analysis and reactive graphics showing a variety of trends including changes of company insider holdings.

111.2 System Diagram



111.3 Actor Identification

In this project, the user is identified as the only actor. The user will have the ability to access different information and data relating to a stock/company — which is dependent on their selection on the Edgar terminal once they have searched for the stock. Insider holdings and fundamentals are two main sets of data the user will have the ability to access. No user account will be necessary, all people have access to the SEC database.

111.4 Design Rationale

1.4.1 Architectural Style

We will use a 3-tier architectural style. We chose this style because we will have a presentation tier for the user interface, an application tier where we analyze and manipulate data, and a data tier where we access data from the SEC database.

1.4.2 Design Pattern(s)

- Adapter
    - converting XBRL from SEC database into JSON (In particular we're converting the GAAP facts from the financial statements into structured JSON which can be used to find values in a way that doesn't require AI or access to expensive API's)
    - This was chosen to standardize the data we receive from the database and make it easier to parse in python. This will allow us to make our charts and display our data to the users more efficiently

- Command:
    - 'Monitor insider holdings' and 'request fundamentals' are two main features in our project and for efficiency we will implement the command design pattern by encapsulating the command request as objects. They will use a display() method.
- Model View Controller:
    - Give graphical view for selected GAAP facts. Could be presenting tables of financial facts, or graphs based on those facts. Data will all be handled on the client side though.

1.4.3 Framework

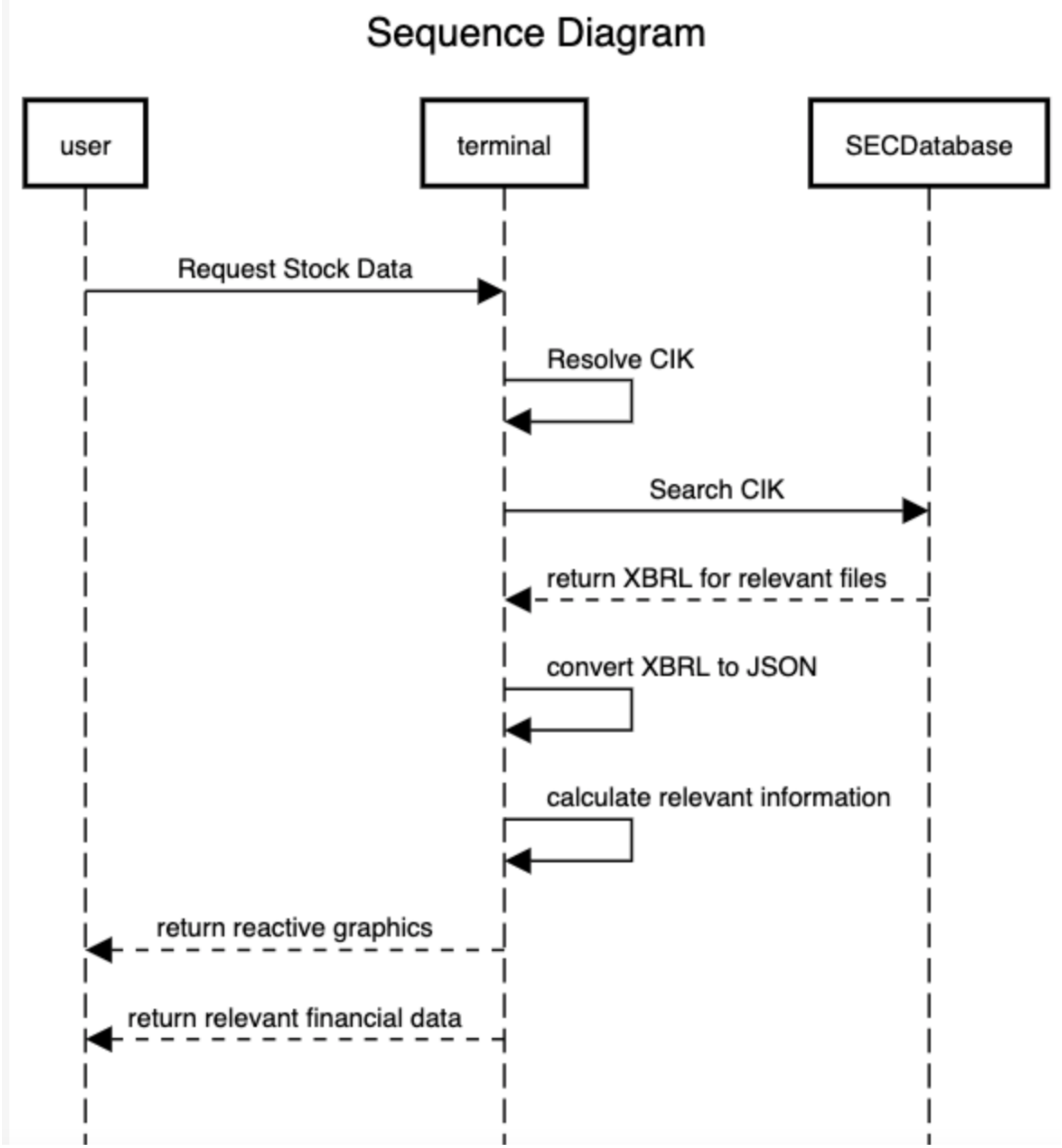
-http (Python 3 library)
-Django (Python 3 web development framework)
-NumPy (Python 3 data manipulation and charting)

The SEC database uses RESTful web services. Once a ticker is resolved by CIK, a query can be created on the fly to request all the necessary data from the SEC, no API token is required.
-Interaction with SEC database can be done through the http library in Python

User Experience can be managed through the Django web framework
-Django is lightweight and simple, enough to satisfy the requirements though.

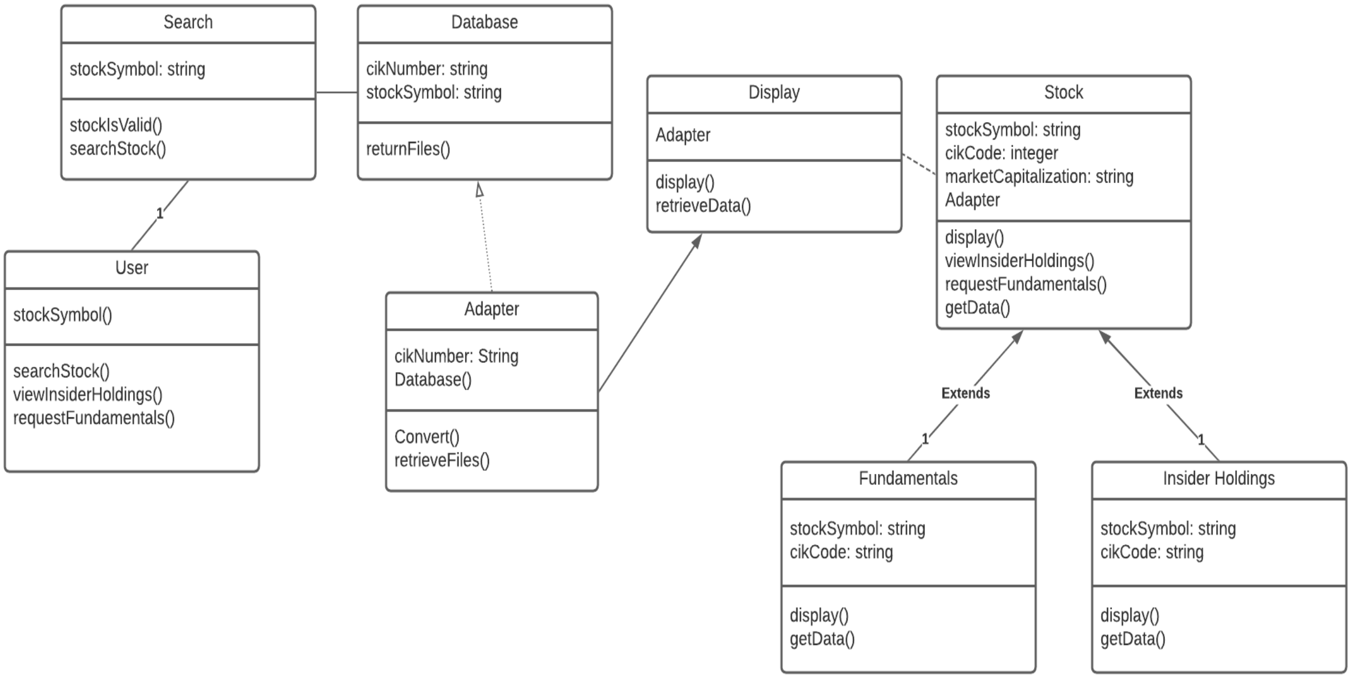
Management of calculations and graphs will be done with Numpy.

112.Functional Design



113.Structural Design

This is the class diagram for the EDGAR terminal


Search
stockSymbol: string
stockIsValid()
searchStock()
Database
cikNumber: string
stockSymbol: string
returnFiles()
Display
Adapter
display()
retrieveData()
Stock
stockSymbol: string
cikCode: integer
marketCapitalization: string
Adapter
display()
viewInsiderHoldings()
requestFundamentals()
getData()
1
User
stockSymbol()
searchStock()
viewInsiderHoldings()
requestFundamentals()
Adapter
cikNumber: String
Database()
Convert()
retrieveFiles()
Extends
Extends
1
1
Fundamentals
stockSymbol: string
cikCode: string
display()
getData()
Insider Holdings
stockSymbol: string
cikCode: string
display()
getData()