**Memo**

To: Professor Pisano

From: Prithika Ganesh [pvganesh@bu.edu](mailto:pvganesh@bu.edu), Ignacio Nunez Gomez [inunezg@bu.edu](mailto:inunezg@bu.edu), Suyash Bhatia [suyashb@bu.edu](mailto:suyashb@bu.edu), Zen Mae Lee leezm@bu.edu

Team: Team 30, DevUp

Date: 03/17/23

Subject: The Art of Valuation: Second Prototype Testing Report

1. **Summary of Equipment and Setup**

1.1 Required Materials

1.2 Set Up

1.3 Testing Procedure

**2.0 Measurements Taken**

2.1 Measurable Criteria

* 1. Measurements Taken Table Results

**3.0 Conclusion**

**1. Summary of Equipment and Setup**

**1.1 Required Materials**

Software:

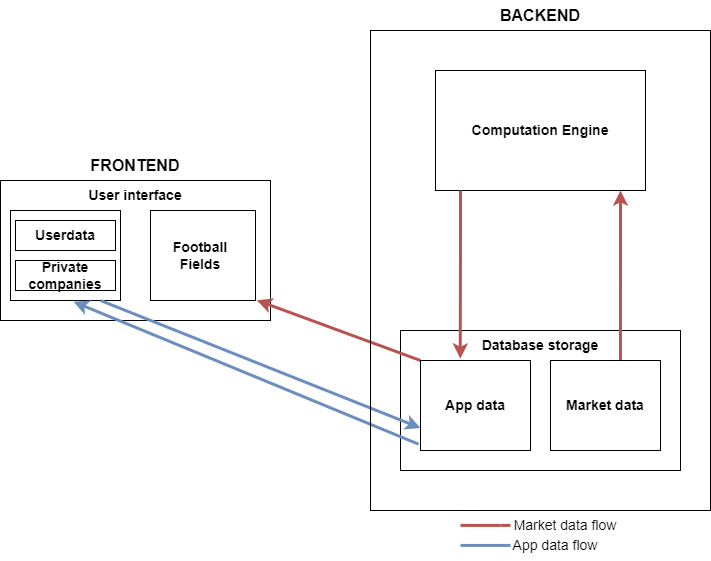
* Emulators: XCode
* Languages: ReactNative and Python 3
* Frameworks: Flask (Python)
* Data APIs and API keys: IEX Cloud Apperate

Hardware:

* Personal computing device i.e. laptop, PC etc.

**1.2 Set Up**

The setup requires running the frontend and backend code on an iOS emulator. The Visual Studio Code IDE will be used as our main platform.



*Figure 1: Illustration of Data Flow from Backend to Frontend*

**1.3 Testing Procedure**

1. For the frontend, open VSC and navigate into the *workshopfinance* directory
2. Run ‘npm start’ → ’i’ for iOS.
3. Run the routes.py file (i.e. python routes.py) in a new terminal to initiate running the backend.
4. Open the IEX Cloud database to demonstrate that our database and tables are designed and functional
5. Open the valuations.py file, and add 3 valuations with the same footballfieldId, to add them to the same football field
6. Show that they have been added to the database
7. Open the emulator, and navigate to the footballFields.js screen, to show how the football field contains the 3 valuations previously created.
8. Change the name of the Football Field
9. Show live changes of Football Field name on database
10. Show changes when you change scale
11. Navigate through the app to show all the created screens

**2. Measurements Taken**

**2.1 Measurable Criteria**

The criteria for successful running and output are as follows:

1. The backend is able to run without any errors. In other words, the file routes.py runs smoothly.
2. IEX Cloud database is designed.
3. IEX Cloud database functional. This can be demonstrated by adding rows to the tables using REST Api.
4. The Football Field name can be changed and automatically updated on the database.
5. The frontend is able to successfully run on the IOS emulator without crashing or any errors.
6. Valuation charts are drawn from data obtained from the database (previously introduced by APIs).
7. Football Field name is changed automatically when the user updates the corresponding user input.
8. The valuation scale changes when the user changes the ‘Scale’ user input.
9. There is a fluent navigation logic across the whole application.

**2.2 Measurements Taken Table Results**

| **Action** | **Category** | **Y/N?** |
| --- | --- | --- |
| Flask working properly (routes.py) running smoothly | Backend | Y |
| IEX Cloud database designed | Backend | Y |
| IEX Cloud database functional: Adding rows to tables using REST Api | Backend | Y |
| Football Field name updated automatically in database | Backend | Y |
| Application runs smoothly on iOS/Android emulator | Frontend | Y |
| Valuation charts drawn with data from database | Frontend | Y |
| Change the name of the Football Field | Frontend | Y |
| Changes in scale when user input is changed | Frontend | Y |
| Navigation across the app flows smoothly | Frontend | Y |
| **Result** | | 100% |

**3. Conclusion**

Our second prototype's main goal was to implement specific features which are integral to our application's core components. These features include backend and database integration, frontend and backend integration, and smooth UX and display of data.

The backend was systematically divided into data gathering/fetching and financial computation. We were successful in implementing both of the components. The financial computation was achieved through a Python script computation engine, and the data gathering/fetching was achieved through the fetching and posting of the necessary data from IEX Cloud to our app’s local database. Since we were successfully able to implement the aforementioned core attributes, we focused on testing smooth, end to end functionality of these functions. We successfully achieved this through displaying how a user’s sign up data was saved into the app’s backend database. Additionally, we demonstrated how the same numeric financial data displayed in the backend was shown through a graph on the frontend of the app. Therefore, we were able to display front and backend integration, successful financial calculations, and data gathering, as the successful valuation creation tests our application’s ability to add data entered by our users at the frontend to our databases at the backend, and financial calculation and display.

On the frontend, we were able to achieve great feats in terms of application development. What was integral for us to complete was smooth functioning of our application on the emulator. Our emulator had previously not performed smoothly, but during our second prototype testing our frontend was able to work on the emulator without any hiccups.

Additional features to our frontend was the ability to successfully draw charts using data supplied from the IEX Cloud database. The frontend was able to retrieve this data and display it on the football field. The frontend is successfully able to edit names of the football fields. This request is sent from the frontend and the backend alters the databases to update them with the new name. Our frontend also is able to successfully “sign up” a new account, having a user’s information entered into our “users” database.

The criteria used to measure the second prototype essentially ensures that these components have no current compilation issues and function smoothly. Integrating the backend and frontend (i.e. displaying data from the backend on the frontend for the user to view) is successfully completed with all features working perfectly. We are now progressively adding small features in order to successfully complete our application. We are anticipating that a successful final application testing will be achieved before our submission.