Design Proposal

Automated Industrial Documentation System (AIDS)

Introduction

The goal of this project is to manage the sales, costs, profit, stocks and trends of corp-wide industrial operations in Eve Online in a way which involves minimal input from the user while still providing the data transparency required to properly evaluate those operations. Industrial operations in Eve require precise monitoring of costs and materials, something which can be time-consuming to do manually. This project is intended to provide real-cost data on current and previous industrial operations, and to record historical trends and profits quickly, accurately and easily.

Proposed Features

This project will be designed to be used alongside Zifrians 'Eve IPH', an application which has served industrialists well for years and continues to excel in its field. Attempting to provide the same features as IPH alongside the current design proposal is entirely un-necessary and would serve only to delay development and over-complicate the scope of this project. It is intended that IPH would be used for determining potential profitability of blueprints as well as generating material requisition lists. With that in mind the proposed features for this project are as follows:

- **Warehouse Management**

  Managing and recording of material stocks is paramount to a successful industrial operation. The warehouse management feature will maintain a virtual stockpile of your materials which would be automatically updated through the Eve AssetList and MarketOrders API. For the initial proposal material costs will be averaged over the entire stock. Using an average price means occasional outliers will be spread over the whole stock, rather than drastically affecting the profits and costs of partial batches. It is noted that this also has the affect of making it difficult to properly capitalise on a significant drop in material costs, as the difference would be spread over your entire stock. However it does not affect the actual profit gained from a drop such as this, and it means material cost spikes are equally normalised.

- **Real-cost Profit Documentation**

  IPH is perfectly adept at determining the potential profitability of blueprints, however it is imperative that actual financial input and output is recorded and analysed. Through monitoring MarketOrders API and using the warehouse feature mentioned above actual costs and profits for all your operations will be displayed. This data will be transparent in nature, so it will be possible to determine exactly how the application has generated the results. This avoids fictional profit data and allows errors to be easily spotted and traced.

- **Historical Performance Data**

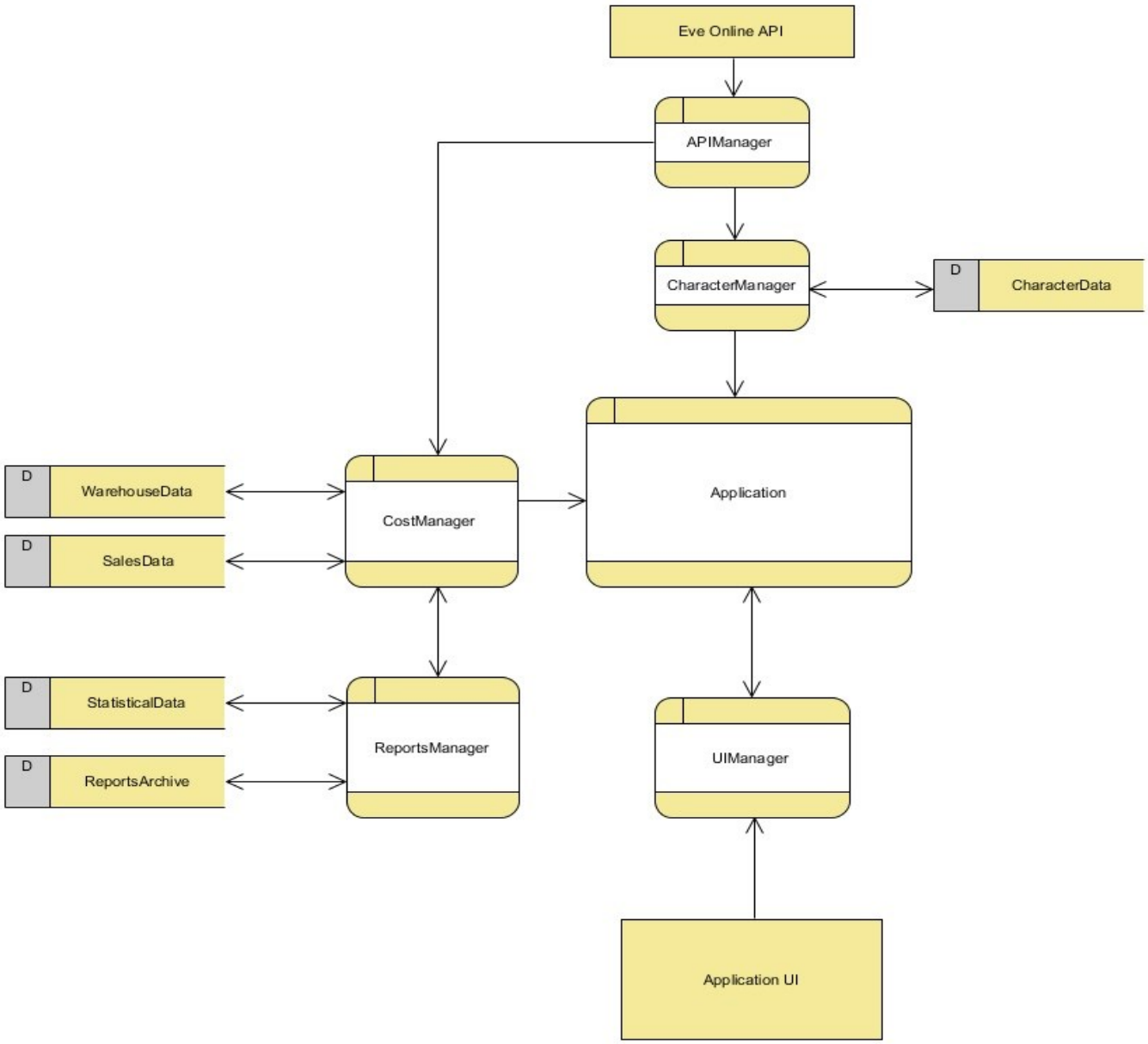
  Although not always imperative to continuing operations, it is valuable to maintain historical performance data. This will be displayed as archived weekly, monthly and annual financial reports with visual aids such as line graphs and pie charts, useful statistics and user-added comments. These reports will have the same transparency as the active week, although they will not be modifiable.

- **Client-side System Structure**

  As this application is designed to be used alongside IPH, it does not require the online database access that it would otherwise. As such everything related to the application will be stored and managed client-side. This obviously has benefits in terms of account and API security. However it also means that system data can be manipulated by the user. As there is no conceivable benefit to this, it is not seen as a major concern. Account data will be able to be archived online through the use of Google Drive, although as an initial proposal this may be something which the user would have to handle manually.

Data Structure

This application will be programmed with Java and use XML for clientside I/O operations. XML is very easy to use and navigate, is human-readable and can read and write Java Class data easily and quickly. Java is a flexible language able to be used on both PC and Mac and is also the primary language of the programmers working with OIRI. The following chart describes initial data structure designs:



Development

Due to the importance of testing in driving the development of this tool, the primary goal will be to complete a working prototype which meets the features proposed in this document. Future development will initially be largely UI and UX based, before any additional features are proposed or worked on. Discussion on ongoing and future development, whether it be new features or modifications to the vanilla features, should take place in the projects' Slack channel, or with the developers in person.

Testing will begin before any semblance of a finished product is available, as it is imperative that issues are caught early on for a clean development. If you are interested in testing any part of this project, or its initial alpha releases, contact 'Ason Osiris' in Eve Online via Eve-Mail or chat.