

4CCS1PPA Group Project Report

Wonjoon Seol, Computer Science with Intelligent Systems, 1600465

March 30, 2017

1 Introduction

In this group assignment we create an application that uses UFO sightings data. This documentation for the application will focus on how the program is used by the users and can be extended to better suit potential users. This will be done by analysing the program by the following ways : Domain modelling, Hierarchical task analysis, virtual windows, and global navigation structure.

2 Domain Model

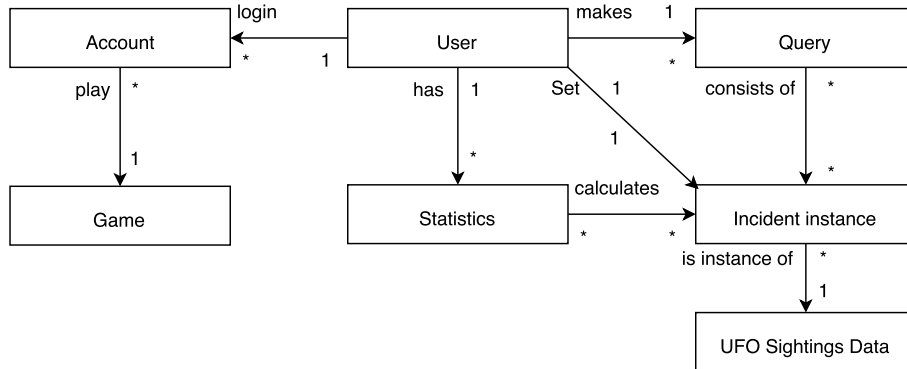


Figure 1: Domain Model

This is our Domain model. Domain modelling allows us to have high level understanding of the program without implementation details or user interface widgets.

The required functionality of the program is not sophisticated. The diagram shows that the user can set the interval for the data incident. The query entity can fetch or sort raw data instances and the statistics entity contains methods to calculate useful statistics. The domain model is able to achieve all tasks required by the brief so no further complex models are necessary. Start date and end date are not included in the diagram as these are better as attributes in 'incident instance' entity.

More interesting points are the cardinality between the user and account. A user might have multiple user account. Either a user wanted to start the game new or simply lost his account. Even though this isn't an ideal situation, it is better to makes the two entities many to many association. This allows the model to at least allow the possibility to have multiple user account. Since the back-end is hardest to change, we need to account for the model to be the as flexible as possible. Same logic applies to multiple accounts being able to play on a game.

Creating the domain model allowed me to have a clear model structure basis to be used to create the back-end functionality.

3 Hierarchical Task Analysis

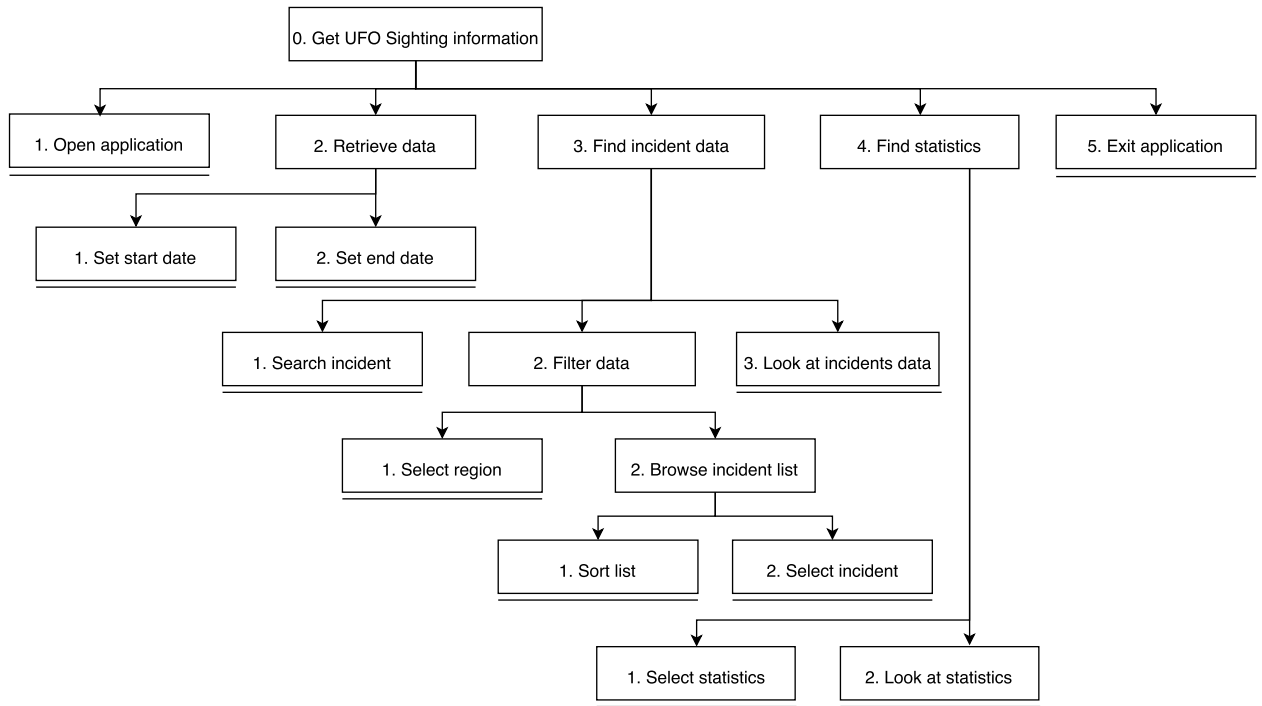


Figure 2: Hierarchical task analysis diagram.

The hierarchical task analyses what user needs to do in order to achieve his intended task. This does not take into account what the program requires to do. By this analysis we can optimise the procedures a user is required to achieve a set goal.

The figure 2 shows user's main task, which is provided in the brief. To get UFO sighting information. The following are the plans required to gather useful data.

Plan 1: Do 1. Then repeatedly do 2, 3, 4. Then 5.

Plan 2: Swap at will between 2.1 and 2.2.

Plan 3: Repeatedly do 3.1 and 3.2, 3.3 until interesting data is found.

Plan 4: Swap at will between 3.2.1 and 3.2.2

Plan 5: Repeatedly do 3.2.2.1 and 3.2.2.2 until happy with the selection

Plan 6: Repeatedly do 4.1 and 4.2.

By using the hierarchical task analysis technique I was able to identify possible extension to improve plan 3, and it is now included in the diagram. The search functionality allows the user to locate the needed incident data without having to go through plan 4 and 5. The plan 4 and 5 provides useful incident listings by categorising and filtering the data by certain types. But if the user already have sound idea of the data, using the search functionality can significantly reduce the time it takes to find useful incident data.

4 Virtual Windows

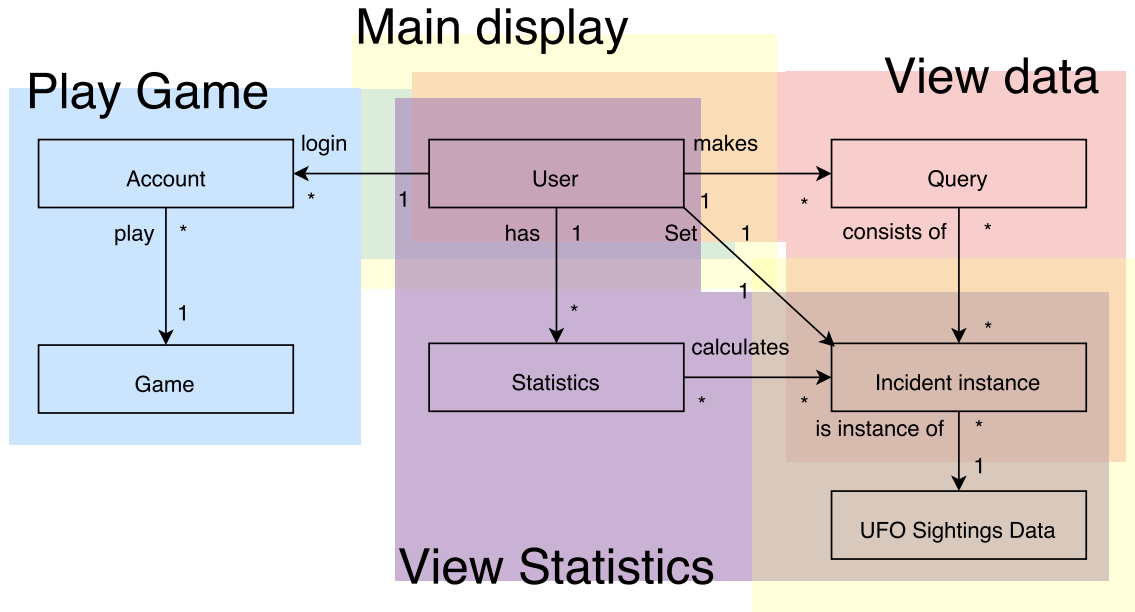


Figure 3: Virtual windows diagram.

The virtual windows link the data model to viewers. The first virtual window, main display window shows loading status of the data instance. This is connected to user, incident instance and UFO sightings data.

The second window, view data represents providing raw data incidents and necessary sorting or filtering methods to the viewer. This window is connected to user, incident instance and query. This window will show alien images on the map, size depending on the incident numbers in the provided period and also provide further information upon the states by if clicked.

The third view statistics window is connected to user, statistics, incident instance and UFO sightings data. The user will be able to view multiple fabricated statistics such as the likeliest state for next UFO report and percentage of non-US to US sightings. The reason for connection to the 'UFO sightings data' entity is due to 'key events in the history' we will do as an additional panel. The window saves current panel choice and loads it again next time the program runs. An improvement can be made by generating separate user save files.

The final windows is the game window, which, if extended, would let the user to log in and continue the game.

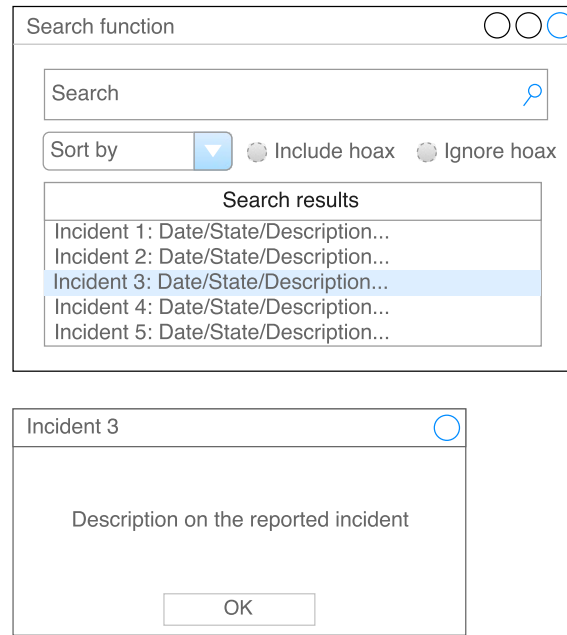


Figure 4: New search function.

This is a mock up of search window to be added as extension. This window will supplement view data window shown in the virtual windows diagram as an extra panel. This window would represents a search method inside Query entity.

In panel two (View data), there will be a search button, which if clicked create a new window. (Figure 4). When the user types any search term including - state, reported date, id and part of description. The query class can parse search query via regular expressions and return appropriate matching data. The window can also contain radio buttons such as 'ignore hoax data' and able to sort the search results if there are many search terms. Once the search results are returned, the user will be able to click the items in the search results to view its full description.

5 Global Navigation Structure

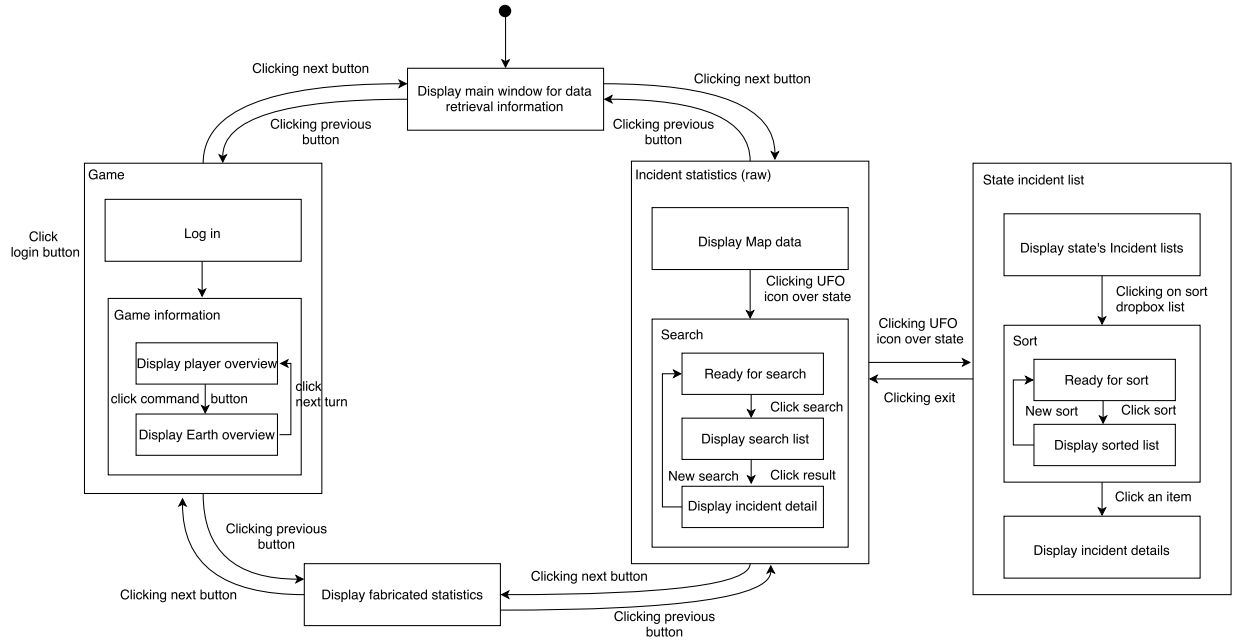


Figure 5: Global navigation structure diagram.

The global navigation structure shows how each windows are connected to another. Using the brief and the virtual windows diagram above I have created the navigation structure (Figure 5).

The user can click left or right button to navigate to next panel. These are wrapped around so that by clicking the next 4 items, it returns to the starting window. Incident statistics (raw) is 'View data' in virtual windows diagram and contains the extended search functionality. Furthermore, the diagram also contains extended game functionality where user can log-in and see some game information.

This diagram illustrates connections between each window and helped me formulating clear understanding of how viewer windows would operate.

6 Conclusion

The domain model, hierarchical task analysis, virtual windows and global navigation structure provide good overview of the application to be created. The benefit of this approach allows the developer to recognise user pattern of the program and be able to extend and improve the original application prior to its actual implementation.

One of the core extension I have added is search function inside view data window. By using the search function the user is possible to skip plan 4 and 5 and retrieve useful data more efficiently.

This report has not covered use cases modelling, so this can be a suitable starting point to move towards to more detailed system design using already established hierarchical task analysis results.