

Name : Rewaz Esmaeel

Student number: 1620327

K-number : 1630551

Email : Rewaz.esmaeel@kcl.ac.uk

4CCS1PRA Coursework 2: Report

Introduction

This report will focus on the NUFORC UFO Sighting Tracker application which gets start year and end year from the user, based on those years a different layout of map and statistics is displayed. The application has many possible extensions which could be made to it in order to improve its features, making it more accessible to both a more casual user, as well as someone using it for a more dedicated purpose.

To allow proper analysis of these proposed changes, a number of analytical techniques shall be applied. The following techniques will be used in order to understand the potentials of the extra feature:

- Domain model
- Hierarchical task analysis
- Virtual window models
- Global navigation structure

The proposed extension is to allow the user to sign up and sign in. Signing in will allow the user to add information about a new sightings that they encounter into the application. This will instantaneously update the data base without having to rely on the external API.

Domain Model:

This will describe the different parts of the application and how they are related.

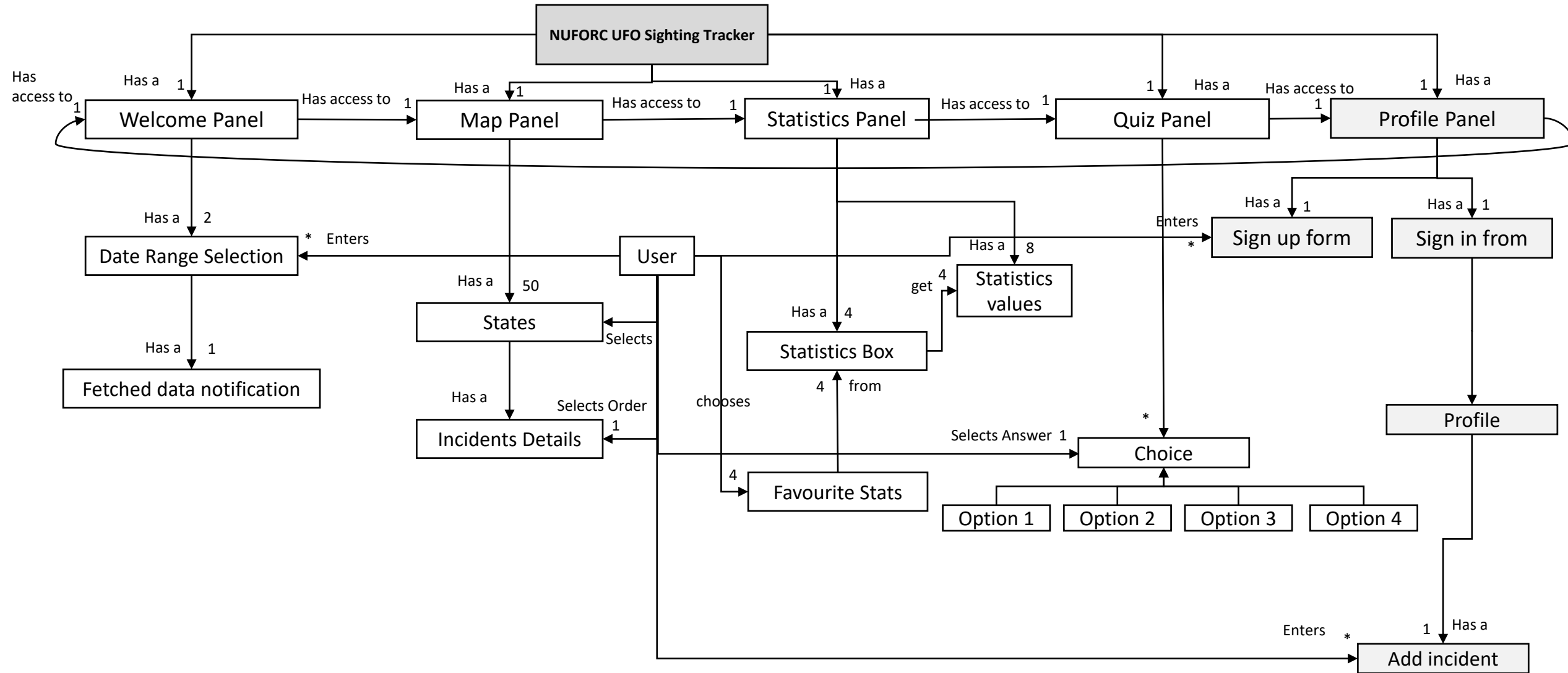


Figure 1: The domain model of the NUFORC UFO Sighting Tracker

The domain of this application involves Sharks, and information about them. The main functionality of the application is to

- search for incidents in a given range of years
- View the incidents in USA map in the given range of years
- View statistics of the incidents in the given range of years
- the ability to keep track of favourite statistics
- Add incidents to all Incidents

Figure 1 shows a representation of the main concepts that were required of the application, including the aforementioned extension which are coloured in grey. The arrows demonstrate how each entity is related, while the 1 and * symbols denote the cardinality of the relationship.

As there may be many users, there is a sign up form which creates profiles for different users.

Different users are able to add details of the UFO sighting that they encounter and adds the details to an existing database of all UFO incidents.

This analytic technique is not efficient as it leads to delays further down the design pipeline, as well as costing additional resources.

Hierarchical Task Analysis:

This analysis represents the order that tasks, and their subtasks, need to be completed during execution of the application. This particular rendition will deal with the starting of the program, navigating to the Add Shark option, adding one, and then exiting the program. This involves identifying tasks and any subtasks they may entail, until there is no more tasks to be undertaken.

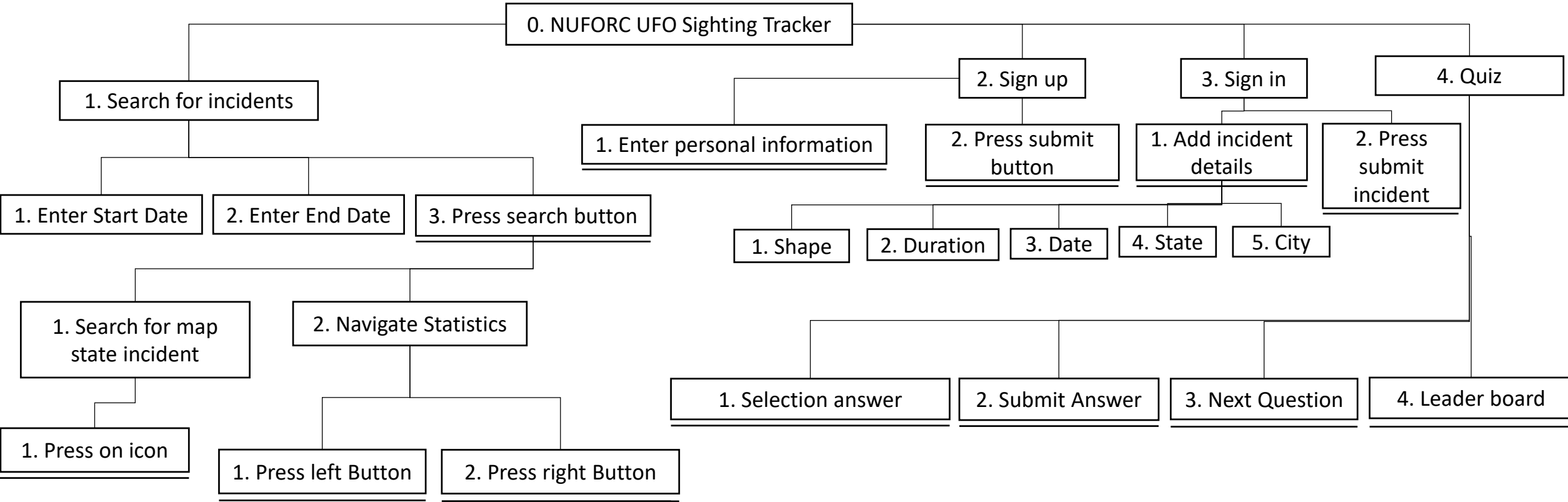


Figure 2: The Hierarchical task analysis of the NUFORC UFO Sighting Tracker

This analysis represents the order that tasks, and their subtasks, need to be completed during execution of the application.

This particular rendition will deal with the starting of the program, navigating to the search incident, Sign up, Sign in and Quiz, and then exiting the program. This involves identifying tasks and any subtasks they may entail, until there is no more tasks to be undertaken.

- The following are plans based on the hierarchical task analysis:
- **Plan 1:** Do 4, 4.1,4.2,4.3,4.4
- **Plan 2:** Do 3.1,3.1.1,3.1.2,3.1.3,3.1.4,3.1.5,3.2
- **Plan 3:** Do 1,1.1,1.2,1.3,1.3.1.,1.3.1.1

Here plan 1 opens the quiz panel then selects answer then submit the answer and then pressed on next question. Then Leader Board is pressed to store the score of the right answers.

Plan 2 opens the sign in and then the incident details are entered which consists of the shape, Duration, date, state and city of the thing that is seen. Then the user presses on submitting the incident.

Plan 3 opens the search for incident where the user enters the start date and the end date of the incident. Then the search button is pressed. Then the users opens the search for map state incident where a state icon is pressed.

Virtual Window Models:

These models will allow a visual representation of which components come together to form different screens. Each different screen/frame will ideally deal with a set of similar tasks, and provide a look at how the user will interact with the program application itself.

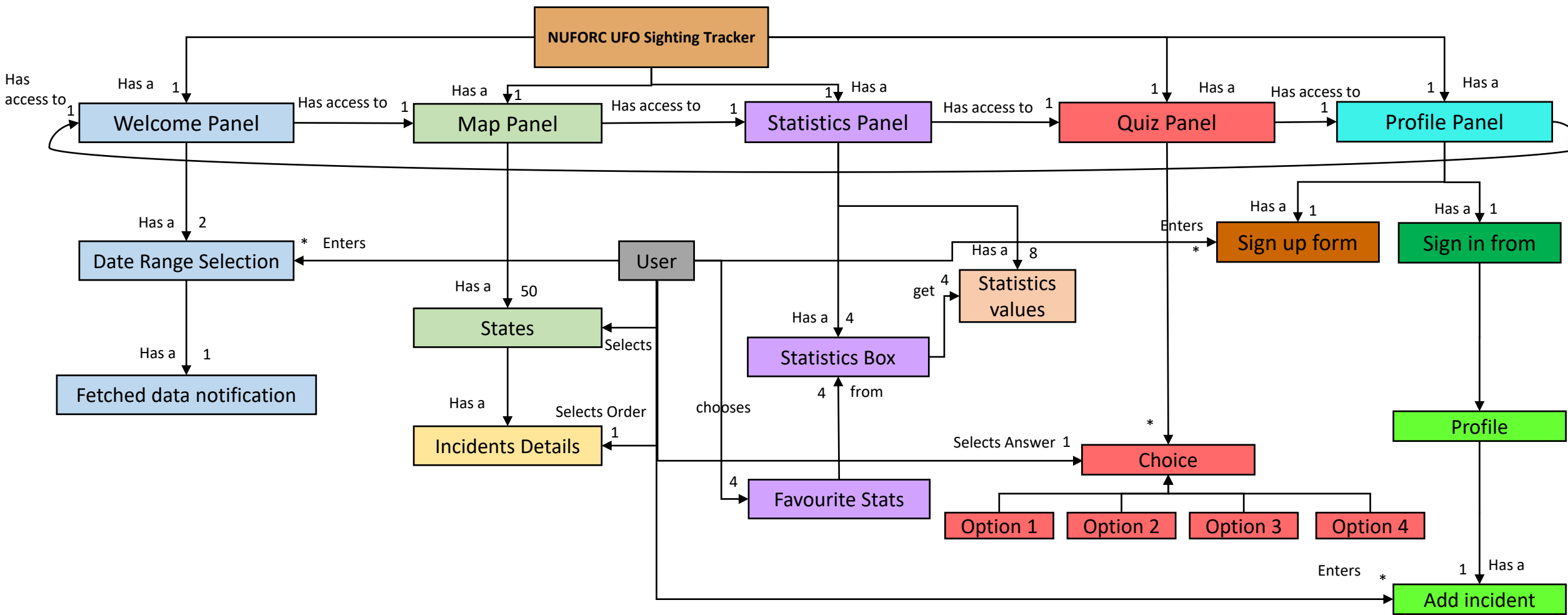


Figure 3: The Virtual window models of the NUFORC UFO Sighting Tracker

Key:

Colour **Teal** for the profile panel where the user has the choice to either sign in or sign up.

colour **rose** is for the quiz panel where the user can test his information and entertain himself while or after the data is being fetched or without fetching any day.

Colour **Lavender** is for the statistics panel where the user can navigate through 8 unique statistics of which are only 4 are displayed. The user can use the navigate right and navigate left buttons to navigate through the statistics. Once the user closes the application his chosen preferences are recorded and loaded when the program fetch data again.

Colour **Grey** is the user who is going to use the application.

Colour **Light Green** is for the user where the incidents are fragmented into USA map depending on the state. And when the user pressed on the icon of the state a new windows shown up which is explained next.

Colour **Light Green** is for the user where the incidents are fragmented into USA map depending on the state. Theses data are shown in a new window.

Colour **Orange** where all the statistics values are stored and are only 4 of them are displayed to the user.

Colour **Blue** this is displayed at the start of the application where the user is informed about the application.

Colour **Brown** where the users can enter their person details to sign up.

Colour **Dark Green** where the user enters the username and password to sign in into their profile.

Colour **Green** is the profile where the user can add incidnets

Using the domain model produced to show how the different potential data types interact, they can now be divided into further possible windows for the extended application (as shown above). Each different term within the domain has been placed within a potential virtual window that will allow it to become a possible data type in the implementation of that window.

This model was generated based on the interaction of the entities within the domain, and builds upon the domain model. Using a basic execution of the application as an example, a typical series of events would be the following:

Upon starting the program, the user would be greeted with the welcome panel (blue), and given the option to navigate through Map Panel (Light green), statistics panel (purple), Quiz panel (Rose) and profile panel (Teal). In the welcome panel the user has the ability to enter start date (from) and end date (to) in order to view the details in the other panels (except the quiz and profile panels).

Global Navigation Structure:

- This section will show how the entire program interacts, describing each section and how they are interconnected. This will illustrate the process of starting the application, and how the user navigates from one part of the program to another.
- This process is heavily based on the virtual model windows, and the behaviour shown therein has been expanded upon to deliver the analysis of the global navigation structure.

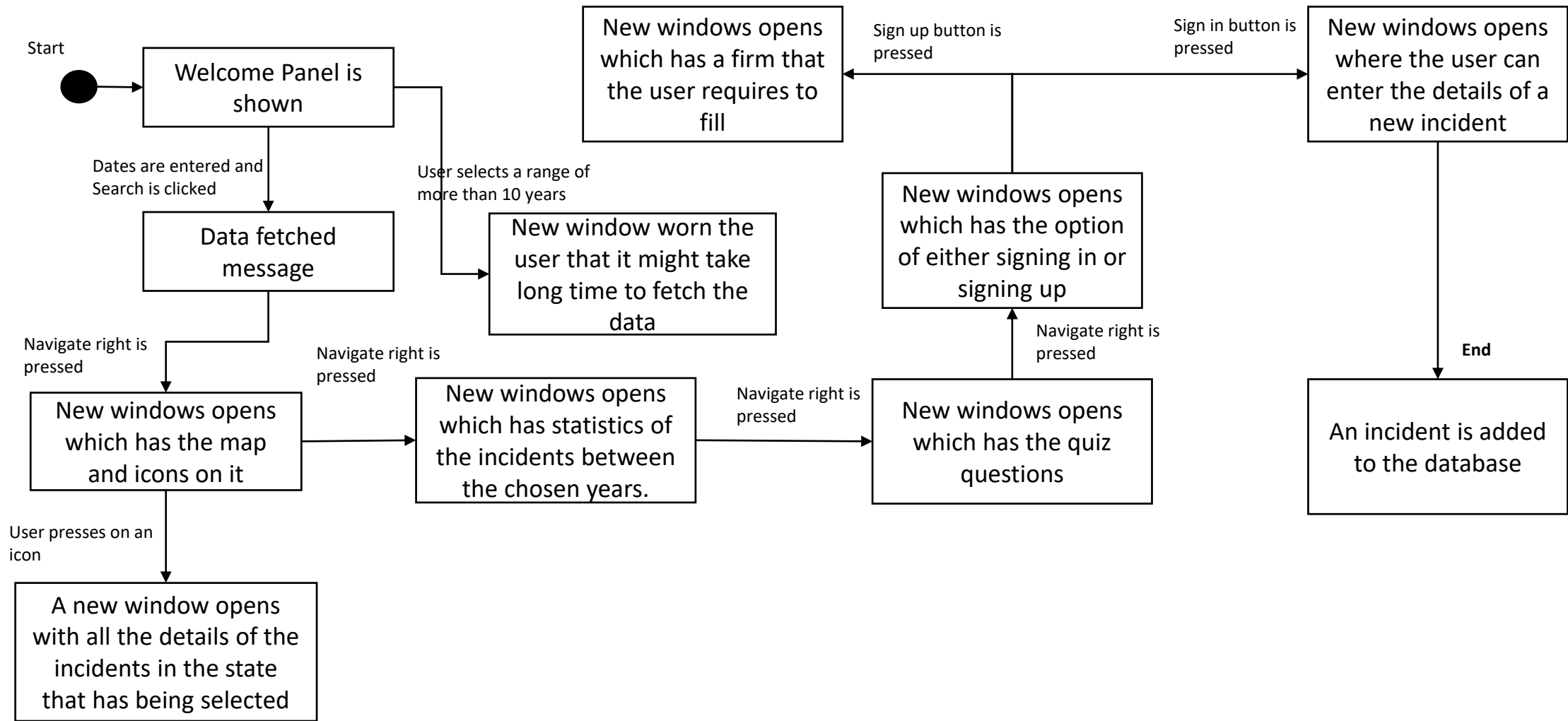


Figure 4: The Global navigation structure of the extension to the NUFORC UFO Sighting Tracker

Each major change in state of the program is shown in a rectangle, and the action which causes this change is shown in text, accompanied by an arrow. The implementation of the planned extension is found in a new frame, for the majority. As shown, moving from one aspect to the program to another is fairly linear and easy to comprehend, something which is very desirable to the end user.

The implementation doesn't interfere with any pre-existing systems, and meets all requirements needed to complete an extension such as this.

If the user selects a data of more than 10 years an warning message is displayed to them notifying that I might take long time to fetch the data.

The buttons of the statistics are not functioning if the user does not select any dates.

To conclude, this global navigation structure provides a clear, succinct implementation of the extensions requirements, where each step is easy to achieve and program flow is obvious.

Summary and Conclusion:

This report aimed to provide a comprehensive, yet brief, overview of the process of implementing an extension to a pre-existing application, while referencing and taking into account the findings of four different analytical techniques.

I have learnt that it is important to talk to a potential user before any actual software development happens, and gather as much data, that can tell us what would users want, as possible. However, equally important is then to process this data and create a research based on it. This planning ensures that the final application will actually be useful to the user and also makes sure that if needed it will be possible and easier to extend it. It also helps to provide better estimate for how long will the application development take to finish.

have also realised that all of these techniques are not without their flaws and limitations. Each technique requires an additional development step, and does not produce any executable code. This effectively leads to in an increase in time, effort and money spent on the procedures.

With regards to actual implementation of the features discussed, it would be a mostly trivial affair. The bulk of the new code would be contained into five new classes extending Jpanel which will be added to the main Frame and in that panel there would be two JButtons where the user has the option to sign up or sign in. If the sign in is pressed the user will get into a new panel where two text fields are displayed and the username and password can be entered. If the user pressed the sign up option then the user need to enter personal details into JTextField the pressing on a submit JButton. We have designed the code in such a way, that if it will be needed, the application can be extended.