Track Attack: Design Document

What Is Track Attack?

Time management is a challenge for many students and workers. Our application aims to assist the user with time management by allowing them to break their work load into categories and tasks, record the time they spend working on each task, review their work habits and be rewarded for spending time on task.

The two canonical users for Track Attack that we thought about throughout the development were:

1. A university student who struggles to organise and manage their workload, and has problems with procrastination. Track Attack can help this student by allowing them to organise and see their work load logically represented, helping them to track their time on task, set goals for time spent on a task, and reminding them of upcoming due dates.

2. A freelance worker who works on many different projects for several clients and needs to keep track of how much time they spent on each project. Track Attack allows for quick and easy clocking in and out, and analysis on the time spent on individual tasks or entire categories.

Our Design Concepts

Track Attack uses the familiar concept of a hierarchical ‘folder and file’ structure to represent tasks and categories. Categories are similar to folders, and tasks are similar to files. We used this interface metaphor because it should be familiar to most computer users.

We also wanted the application to feel small and lightweight, to encourage the user to keep it running while they worked. For this reason, we limited ourselves to a small window size, and built in a ‘Lite’ mode to minimise the use of screen real estate.

Flexibility and interface ‘forgiveness’ was another important concept in our design, and so we built in functionality to search, edit, move, and delete tasks and categories.

We wanted Track Attack to be enjoyable to use, in contrast to some other time tracking software that feels boring and corporate. To aid with this we put time into creating images for each button and achievement. We also focused on small details such as progress bars, clock in/out animations, tooltips and color coordination to make using the application into an enjoyable experience.

Implementation Details

Track Attack is a desktop application and was built using JavaFX, Jfoenix and MySQL.

When the program starts, we load in saved tasks, categories and their associated data from the database. All of this back end information is loaded into a group of backend classes defined in the src/model folder. The interaction between the front end, back end, and database all happens through the InterfaceDriver class defined in InterfaceDriver.java. This class contains the hierarchical category/task structure, and functions to access and modify this structure.

We took this architectural approach to allow us to work independently on the back end, front end and database without stepping on each others toes. Additionally, accessing and changing all data through one interface meant that changes to the back end didn’t accidently affect the front end.

The front end has two main parent classes, Controller and Screen. Each screen in the application has a controller class which inherits from Controller, a screen class which inherits from Screen, and a .fxml file which defines the layout of objects on the screen. The inheritance allowed us to easily change screens by passing around screen objects and polymorphically calling start() to run a new screen. The controller then handled all of the behaviour of the screen.

The codebase is a little messy, with commented out code and some duplication, but the underlying architecture served us very well and we feel that the application is easily extensible and maintainable in its current state.

Further development might consist of porting the application to Android or iOS to allow for mobile use (made easier because of our clean front end/back end split), improving analysis capabilities, and allowing for online synchronisation of tasks between several devices.