Time Tracker

Plan your future based on what worked for you in the past.
#Ultimate-Scheduling-Assistant

Front End
User Interface
Written in HTML, CSS, JavaScript

Drivers
Connection between data and UI
Written in C++

Back End
Database
Written in SQL

- Enter and Save new Activity
  - Did projects get done on time
  - Was this activity stressful
- Navigate through charts displaying previously saved activities
- Displays the best weeks/months in order of most productive/stress-free to least productive/stressful.

- Connects to database
- Accepts input from UI and transfers it as output to the database into the proper tables
- Accepts requests from the UI and gathers data from the database as input, presents it to the UI so that it can display accurate charts based on the users data.

- Store multiple datasets based on relevant activity attributes
- Query list must include data necessary to populate chart options in the UI

Front End
User Interface
Written in HTML, CSS, JavaScript

Main Menu
* Navigation bar
    - Create new activity
    - Store new activity
    - Indoor outdoor chart
    - Work leisure chart
    - With people vs alone chart
    (more options can be added)
    (options can be adjusted as needed)

Create new activity window
* same navigation bar
* New activity fields
    - Activity name
    - work/leisure
    - indoor/outdoor
    (additional fields as needed)
*submit button

Store activity window
* same navigation bar
* Name of activity you are storing
* list of people you were with
    - list can be check boxes etc
* Time spent doing activity
* Activity date
* Stress level after activity (scale 1 to 10)
* Percent of activity that was completed on time (scale 0 to 100%)
*Store activity button

Chart window
*same navigation bar
* Toggle chart type holding whichever data was selected

Chart types
- PI chart
- Vertical Bar chart
- Horizontal Bar chart
(Whichever charts we decide to use for the final product)

Request data from DRIVER to be retrieved from DATABASE

Send data to DRIVER to be stored in DATABASE

Driver
Connection between data and UI
Written in C++

UI connector class

Chart type class

Derived classes:
Indoor outdoor chart
Work vs leisure chart
With people vs alone chart
(other defined charts)

Separate data type prior to sending it to database

Database Connector class

I/O to and from DATABASE and DRIVER

Back End
Database
Written in SQL

Storing Data Queries

Retrieving Data Queries
(Incl. Table Views)

Tables

Activities

Act_num -PrimKey
Act_name
Leisure/work T/F
Inside/outside T/F
New_Skill T/F
Old_Skill T/F
RepeT T/F

People

Person_num –
PrimKey
Person_group
(fam,friend,etc.)
Person_name

Activities_History

Hist_num –
PrimKey
Activity_num
Hist_date
Start_time
End_time

People_History

Connector Table:
(no PrimKey)

Person_num
History_num