# *Report Sections*

## Analysis

* **Introduction**
  + Parents’ Evening: A short scheduled meeting or conferences between the parents and teachers of students to discuss both a child’s progress at school and find solutions to academic or behavioural problems. Since teachers will have dozens of students to interact with, a scheduling system must be implemented in order to allocate specific time slots to students, parents and teachers to reduce clutter, confusion and delays.
* **Requirements**
  + Assign an ID as well as a parent’s information (incl. student information).
  + *Receive one’s preferred day and time frame during the 3-day Parents’ Evening period.*
  + *Display the available time slots and fill them in on a first-come-first-serve basis.*
    - *Create logic to accommodate for conflicting time slots.*
  + *Allow the parent(s) to both delete and/or reallocate their scheduled meeting.*
  + *Provide a menu displaying multiple actions that the user can perform from a home page.*
  + *Output the final scheduled timetable to the parent/teacher.*
  + *Save any edited data to both the ‘parent’ and ‘schedule’ JSON files.*
* **Sub-problems** [**Decomposition:** The process of breaking down a large problem down into smaller, more manageable parts which are easier to solve.]
  1. Parent information
     + To uniquely identify each parent, this sub-problem should both verify and allow parents to register new information.
  2. Time allocation (inputs)
     + The main function of the program: receive a preferred day and time frame from the user; display available slots within the preference (or bordering available time slots if the former is unavailable); confirm the selection; convert to an object and store the data.
  3. Displaying the schedule
     + By incorporating a table-formatting library, display the schedule for both the teacher (one which may span the 3-day period) and the parent (one which may only span 1 day with their time slot included) depending on who is accessing the program.
  4. Cancellations and reallocations
     + Re-display the user’s selection and ask for confirmation to delete the time slot. If they choose to also reallocate, continue the function and link it back to the time allocation function.
  5. Exiting and saving the program (data)
     + Force quit the program when the user has completed their desired actions and write both their user information and scheduling data to a (JSON) file.
  6. Menu
     + Neatly formatted menu to allow the user to select an index to be redirected to the other functions performing different actions.
  7. Main program – information
     + Initial start-up of the program: introductory information to initiate the program.

## Design

* **Solution Design Algorithms**
* **Individual Subroutine Plans**
* **Initial Test Plan**

## Implementation

* **Program Code**
* **Screenshots of Debugging Skills to Correct Errors + Algorithm Refinements and Justifications**

## Testing

* **Completed Test Plan**

## Evaluation

* **Evaluation and Reflection of Key Requirements of the Program**

## References

* **Any sources of information you used to complete the project. – Name of source/Date of access**