**Motivation**

Our client is the owner of several daycares across the United States. The age of the children can vary from six weeks old to six years old. By law, the daycares must follow strict child to teacher ratios for various age groups of children. Children quickly move from one age group to another and require constant reassignment to different classes. Exceptions can be made to hold back a child if the child has not met requirements necessary to move to a higher class. There are a limited number of teachers and classes available.

Our client is normally very close to full occupancy with many of his daycares having waiting lists of over 100 children. Priority is given to children who already have siblings enrolled in the daycare and to children whose parents are in the teaching profession. A lower priority is given to children that have early deposits, and the lowest priority is given to children with no deposit.

Currently our client has one of his staff manually entering the dates when the children move from one class to another. This procedure initially was not time consuming but as the daycare grew to well over 600 students, this job became very taxing to do manually.

When a new child is accepted into the daycare he or she should be guaranteed a spot in each class as the child outgrows one class and moves to another. This restriction put further strain on the manual process by requiring the staff to not only make sure a spot is open for the child currently but also make sure a spot is open for all classes for at least 2 years into the future.

Our goal is to minimize the time it takes for the client to schedule and maintain the appropriate classroom occupancy. We will achieve this goal by automating the tedious manual process of scheduling classes while maintaining appropriate child to teacher ratios.

The client and members of his staff will provide us with feedback from time to time as the project advances. The client has already provided the basic functionality requirements that the project should address. If time permits, the client may request additional functionality that was not originally planned in order to better address the client’s needs.

**Functionality**

The most important functionality the client is looking for is, the ability to input a child's information and receive automatic scheduling information. Additional primary functionality includes:

1. Exceptions – Children can be removed from the automated process and inserted manually into the system.
2. Age Group – Client will be able to define the various age groups used by the daycare.
3. Classroom Criteria – Client will be able to define the size of each classroom and the child to teacher ratio for each age group.

Secondary functionality is prioritized as follows:

1. Waitlist Enrollment – the software will determine when children from the waitlist would be able to enroll in the daycare.
2. Waitlist Prioritization – Children in the waitlist will be prioritized for enrollment
3. Statistical Information – Statistics that aid in forecasting information such as child turnovers, parental occupation, waitlist size, etc. will be provided.

**Implementation**

Currently the client uses Microsoft Excel to store this information and the client is familiar with Microsoft Office products. To provide familiarity to the client, Microsoft Access will be used to store and present data for the client. A software tool written in either C or Java will be used to provide the needed functionality the client desires. A user interface will be created to enter children information in an easy to use fashion and provide useful information to the client.

This project will be achieved in iterations. Tentatively, these will be:

1. Create software to model the problem accurately.

2. Create optimal scheduling using existing linear program APIs.

3. Create interface to allow for additional required functionality for the client.

4. Allow for client specified file formats, etc.

5. Bonus features (such as email alerts, additional configurations, additions to modeling flexibility, etc.)

These are subject to change pending design, development, and any feedback received from the client.

**Technical Challenges**

1. Due to the sensitive nature of the data being handled, the software must be error-free, any errors can be very costly to the client. This requires sound computer science strategies and techniques to be used in order for the project to be a success.
2. The Scheduling of children in the daycare is complex and will require good programming skills in order to solve. This problem’s scheduling solution will be explored by employing techniques from linear programming in order to find the most optimal solutions.
3. It is possible that not all of the secondary functionalities will be implemented before the project is over. The software should be upgradable to allow more functionality to be added in the future. Software techniques that allow such upgradability should be used such as object orientated design. We will be following agile development practices (scrum) to achieve this goal: <http://braintrustgroup.com/assets/2013/10/Scrum-Diagram-Thumbnail-01.jpg>

**Team Members**

1. Keivan Sabet is a computer science graduate student with a strong background in mathematics and science. Keivan has interests in creating efficient and easy to use software.
2. Jonathan Brown - currently studying computer science and mathematics. Interests in modeling, automation, and creating efficient, scalable software.