Project Description

A *Pokedex* is a searchable database of Pokemon, listing information about each species. Over the past quaerter-century, many Pokemon fans have chosen to express preference for their favorite Pokemon species by purchasing merchandise related to that species. Because there are so many Pokemon, though, it is not practical for retailers to stock products depicting every species. Some people believe that this limitation could best be overcome by a software system that allows users to design and order their own products featuring their favorite Pokemon, and this proposal describes such a system. For an initial release, the goal is to partner with a sneaker manufacturer to offer variations of their existing sneaker designs in new colors.

These are the primary requirements for the proposed software product:

1. As a Pokemon fan, I want to order quality sneakers that clearly depict my favorite Pokemon.
2. As a Pokemon fan, I want the ability to preview the sneakers I design and see sizing and price information for the design I've made.
3. As a Pokemon trademark holder, I want to monetize the intellectual property of less-popular Pokemon by allowing customers to order merchandise featuring that Pokemon.
4. As a sneaker manufacturer, I want to increase revenue and profit by targeting a new potential customer base.
5. As an investor in the development project, I want the product to leverage existing services so that new development effort is not required to design sneakers when new species are introduced in the future.

The recommended Pokedex service provides an SDK with the interface described below. The sneaker manufacturer's existing website defines a web API for making web requests to order sneakers with the selected parameters, and there is a compatible SDK available for that API, also described below.

Pokedex.h

```
#include <string>
using namespace std;

// Possible values for the Pokemon's type(s)
enum pokemonType {Normal, Fire, Water, Grass, Electric,
Ice, Fighting, Poison, Ground, Flying, Psychic, Bug,
Rock, Ghost, Dragon, Dark, Steel, Fairy, None};

// Each instance of this struct represents one species
// in the Pokedex.
struct Pokemon {
int id;  // Unique number for this species
string name;
pokemonType firstType;  // Must NOT be "None" type
pokemonType secondType;  // May be "None" type
int** photo;  // Bitmap of an image of the Pokemon
// Other fields ommitted

// Returns an array of strings that each represent
// an RGB hex value that was found in the photo,
// ordered from the color that appeared most to
// the color that appeared least. The array always
    // returns at least two colors, and is terminated
// with an empty string indicating the end.
string* getColorDistribution();
};

// Returns struct for Pokemon with this unique id.
// If the id is invalid, returns a struct with id 0;
Pokemon getPokemonById(int id);

// Returns a Pokemon struct for Pokemon with this
// species name. If the name is not an exact match,
// returns a struct with id 0;
Pokemon getPokemonByName(string name);

// Returns an array of Pokemon matching the search
// term, ordered by the field described. An empty
// query will return the entire Pokedex. The array
// is always terminated with a Pokemon with id "0".
Pokemon* searchPokemon(string query, string sortOrder);
```

Sneaker.h

```
// Each instance represents a type of fabric or other
// substance from which the shoe is constructed.
```

```
struct material {
string type; // "Leather", "Rubber", etc
string color; // RGB hex value
};

// Represents a particular design with the selected options.
struct sneaker {
// Name of the pattern used by the manufacturer.
string model;
material soleMaterial;
material exteriorMaterial;
material interiorMaterial;
material tongueMaterial;
string shoelaceColor;
// Bitmap of RGB values to be displayed on the back.
int** heelLogo;
// Bitmap of RGB values to be shown on the tongue.
int** tongueLogo;
float size;

// Returns price in US dollars for selected options
float getPrice();

// Returns a bitmap image showing a digital
// representation of what the sneaker would look like
// from various angles with the selected options.
int** getRender();

// Confirms that a sneaker with the selected options
// can be ordered. Returns "true" if this sneaker can
// be ordered as-is, and "false" if not.
bool getAvailability();

// Places an order for a pair of sneakers with the
// selected options, to be paid for with the given credit
// card number, and delivered to the given delivery
// address. Returns "true" if the payment succeeded
// and the sneaker can be delivered to the given address.
bool order(int creditCardNumber, string deliveryAddress);
};
```