Specification for *Record of Records!* Version 1.00

Purpose

This application allows information about a person's musical collection to be stored. It can store information about individual tapes, records, CDs, and so forth. In addition, it allows information to be stored on the musical groups contained within the music collection.

Objectives/Capabilities

1. Be capable of storing information on which storage medium a particular musical item is contained, including cassettes, albums, CDs, and DATs. In addition, the system should allow for other types of mediums to be used.
2. Be capable of storing information concerning the cost and value of each musical item. In addition, the system should provide a means for storing other identifiable information. See data matrix later in specification.
3. Be capable of reporting the information that has been provided. These reports are defined in their own specification.
4. Be capable of tracking information about the musical groups contained within the musical items. This information should include the members of the group, along with the albums they have produced. See data matrix later in specification for specific information about musical groups that can be tracked.
5. Individual songs on an album should be tracked along with their duration. The number of songs for a single album should be unlimited. (Actual maximum of 250 songs.) This information is detailed in the data matrix later in this specification.
6. There should be the capability to add new musical items, change information on existing musical items, or delete musical items.
7. There must be the capability to add, change, or delete storage medium types.
8. There must be the capability to add, change, or delete musical group information.
9. The capability to see the information on a musical group should be available when entering musical items.

Rules/Assumptions

1. Data will be stored on Groups, Musical Items, and storage mediums separately.
2. If a musical item is added that does not have a corresponding musical group, a warning should be given.
3. If a musical group is deleted that has any corresponding musical items, a warning should be presented.
4. The capability to automatically go from one album to another should exist.
5. Musical items should be sorted by title when going from one musical item to the next.
6. (Technical) Will need to be able to rebuild the databases in case of corruption.
7. The word *album* may be used to signify any musical item.

Processing Components

```
SCREEN:  MUSICAL ITEMS
Prototype:
                          Musical Items                      00/00/00
   File    Edit    Search   Help


     Title:  ----------------------------
     Group:  ------------------------

     Medium:  --

     Date Purchased:  99/99/99     Cost:  $---.--     Value:  $---.--

         Track   Song Title                                  Time

         01:     ---------------------------------------     --:--
         02:     ---------------------------------------     --:--
         03:     ---------------------------------------     --:--
         04:     ---------------------------------------     --:--
         05:     ---------------------------------------     --:--
         06:     ---------------------------------------     --:--
         07:     ---------------------------------------     --:--

                                        Total Album Time: --:--:--


<PgDn=More Titles>                                  <F10=Action Bar>
```

Keys/Functions

Initial Entry

Upon initial entry to the screen, all fields will be blank. The cursor will be on the Title field.

<Enter>

<Tab>

Move the cursor from one field to the next. If the cursor is on the last field, it should move to the first field.

<Shift><Tab>

Move the cursor from one field to the previous field. If the cursor is on the first field, it should be moved to the last field.

<PgUp>

Place the cursor on the first field on the screen.

<PgDn>

Place the cursor on the last field on the screen.

<F1>

Display box containing information on using the current screen.

<Esc>

<F3>

Enable the users to leave the screen. Before leaving, the users will be prompted to see if they are sure.

<F4>

Before adding the record, all edits within the data matrix must be passed. The edits should be evaluated starting with the first field on the screen and working to the last field. If an edit does not pass, then an error message should be displayed and the cursor should be returned to the field in error.

If all the edits pass, prompt the user to add the record to the file. If the user agrees, add the record, and display a message. After the user responds to the message, the screen should be re-initialized.

<F5>

If the user is adding a new record, display an error stating the record must be added.

Before updating the record, all edits within the data matrix must be passed. The edits should be evaluated starting with the first field on the screen and working to the last field. If an edit does not pass, then an error message should be displayed and the cursor should be returned to the field in error.

If all the edits pass, prompt the user to update the record in the file. If the user agrees, update the record, and display a message. After the user responds to the message, the screen should be re-initialized.

<F6>

If the user is adding a new record, display an error stating the record cannot be deleted because it has not been added.

Prompt the user to make sure he wants to delete the record. If the user agrees, delete the record, and display a message. After the user responds to the message, the screen should be re-initialized.

<F7>

Display a message briefly that states that the previous record will be displayed. The previous record in the database should then be displayed. If there is not a previous record, display an error message.

<F8>

Display a message briefly that states that the next record will be displayed. The next record in the database should then be displayed. If there is not a next record, display an error message.

<F10>

If the cursor is on the entry screen, place it on the first item on the action bar.

<ALT><S>

Prompt the user with the search dialog box described later in the specificaiton.

Action Bar Items

<FILE>

<Exit>
Same function as <F3>.

<EDIT>

<Add>
Same function as <F4> except that user should not be prompted to add.

<Change>
Same function as <F5> except that user should not be prompted to update.

<Delete>
Same functions as <F6>. User should be prompted to delete.

<SEARCH>

<Find>
A dialog box should be displayed prompting the user for a Title. The first record found with a matching title should be displayed.

<Next Record>
Same function as <F7>.

<Previous Record>
Same function as <F8>.

<HELP>

<Help>
Same function as <F1>.

<About>
Display a dialog box with information on the program.

Special Musical Item Functions

<Alt><G>

Pop up a box that dispays the description field from the group file. If a matching group does not exist, display an error.

Total Time

This field should keep a running total of the times entered in each Song Title. It should originally default to 00:00:00.

```
SCREEN:  GROUPS
Prototype:
                                  Groups                          00/00/@00
   File    Edit    Search   Help


     Group:  -------------------------

     Date formed:  99/99/99

     Type of Music:  XXXXXXXXXXXXXXXX

     Members:

        ------------------------------------------------------------
        ------------------------------------------------------------
        ------------------------------------------------------------

     Description:

        ------------------------------------------------------------
        ------------------------------------------------------------
        ------------------------------------------------------------


   <F1=Help>   <F3=Exit>                                <F10=Action Bar>
```

Keys/Functions

This screen should operate in the same manner as the Musical Items screen with the following exceptions:

<Alt><S>

Instead of searching for a Title, prompt and search for a group.

```
SCREEN:  MEDIUM TYPES
Prototype:
                                  Medium                          00/00/00
   File    Edit    Search   Help
```

```
Medium Code:    ---

Medium Description:  ----------------------------
```

```
<F1=Help>    <F3=Exit>                              <F10=Action Bar>
```

Database Access Requirements

Musical Items—sorted A to Z by title

- ☐ Need to be able to add new title.
- ☐ Need to be able to randomly read existing title information.
- ☐ Need to be able to sequentially read existing title information.
- ☐ Need to be able to delete an existing title.
- ☐ Need to be able to update information on a specific title.

Groups—sorted A to Z by group name

- ☐ Need to be able to add new groups.
- ☐ Need to be able to randomly read existing groups.
- ☐ Need to be able to sequentially read groups.
- ☐ Need to be able to delete a given group.
- ☐ Need to be able to update information on a given group.

Medium Types

- ☐ Need to be able to add new medium type.
- ☐ Need to be able to randomly read existing medium type.
- ☐ Need to be able to delete a given medium type.
- ☐ Need to be able to update information on a given medium type.

Table Specifications 1.1. ALBUMS Matrix.

Field	Type	Size	Edits
Title	Alphanumeric	30	Cannot be blank.
Group	Alphanumeric	25	Cannot be blank; Give warning if not in Group file.
Cost	Alpha	5.2	Must be positive number or blank (999.99).
Value	Alphanumeric	5.2	Must be a positive number or blank (999.99).
Date purchased	Alphanumeric	8	Must be a valid date or blank. If filled, must be before today.
Medium type	Alphanumeric	2	Must be a code in the Medium file or blank.
Number of songs	Alpha	3	Cannot be greater than 250.
Total time	Numeric	No edits	Protected.
The following fields occur multple times depending on the value of "Number of songs."			
Song title	Alphanumeric	40	Cannot be blank if length*continues* filled in.
Song length (minutes)	Numeric	2	Cannot be negative.

Table Specifications 1.1. continued

Song length (seconds)	Numeric	2	Must be from 0 to 59 if not blank.

Table Specifications 1.2. Medium Types Matrix.

Field	Type	Size	Edits
Medium code	Alphanumeric	2	Cannot be blank; Must be unique.
Medium description	Alphanumeric	35	Cannot be blank.

Table Specifications 1.3. Groups Matrix.

Field	Type	Size	Edits
Group Name	Alphanumeric	25	Cannot be blank.
Date formed	Date	8	Must be valid date or blank; If filled, must be before today's date.
Description of group	Alphanumeric	???	No edits.
Type of Music	Alphanumeric	20	Must be one of a set list to be determined later.
Members[6]	Alphanumeric	30	No edits.

Technical/Implementation Notes

1. Complete prototypes have been developed in a Demo II file called ALBUMS.DBD.


SAMS
PUBLISHING