## 2.1 Product Overview

Manage 2000 is designed to provide project managers with an effective, efficient way to address problems regarding staff placement. In order to achieve this solution, Manage 2000 intelligently surveys individual employee traits, provides staff recommendations based on project position requirements, and dynamically updates database information as personnel complete tasks. Additionally, Manage 2000 opens a more effective channel of communication with upper-level management. For instance, project leaders are obligated to routinely report project progress and or complications to upper-level management. While it is necessary for a project leader to have a good understanding of the various details regarding project staff, upper-level management usually maintain a less significant role with respect to specific project information. To help bridge this information gap between project level staff and higher management, Manage 2000's structure permits information about a specific project to quickly be retrieved and relayed to upper management. Manage 2000 executes this by storing information about positions available within a project with regard to skills considered necessary to complete tasks delegated to each position.

As a result of the staff tracking abilities, Manage 2000 yields yet another benefit without the system having to perform any additional work.

While submitting bids or contract proposals, many companies feel compelled to validate that they retain the necessary skills within their employee base to accomplish the project. As opposed to gathering resume material from each employee, Manage 2000 already contains the essential information about potential project workers. Consequently, Manage 2000 provides instant knowledge about employees and the skills that each holds. Due to dynamic refresh module, information stored within Manage 2000 will be the most up to date information available about each employee. As well as assisting larger corporate organizations, the dynamic refresh module appeals to associations which maintain less staff. For this reason, Manage 2000 inherently increases its own marketability.

In summation, Manage 2000's design will assist project managers by performing the following:

- Tracking employee statistics relevant to project related domains
- Analyzing database records with supplied input and providing staff placement recommendations

- Dynamically refreshing employee database information upon task completion
- Providing an employee skills index corresponding with furnished project requirements
- Preparing resume affiliated employee information and statistics

**2.2 Internal Data Management Module**

The internal data which must be stored during runtime is both large and complex.  Information regarding the current project must be managed: current positions, skills required for these positions, and current personnel assigned to these slots must also be maintained.

Milestone: Functional Internal Data Management Module

Subtasks and schedule:

- Identify data types based on functional specs documentation.  Time: 1 week.
- Create structure / class storage type.
  - -Create / Delete new persons.  Time: 3 days.
  - -Edit loaded personnel.  Time 3 days.
  - -Create / Delete skills.  Time 3 days.

- -Edit loaded skills.  Time 3 days.

- Data type interface (information exchange modules).  Time 4 days.

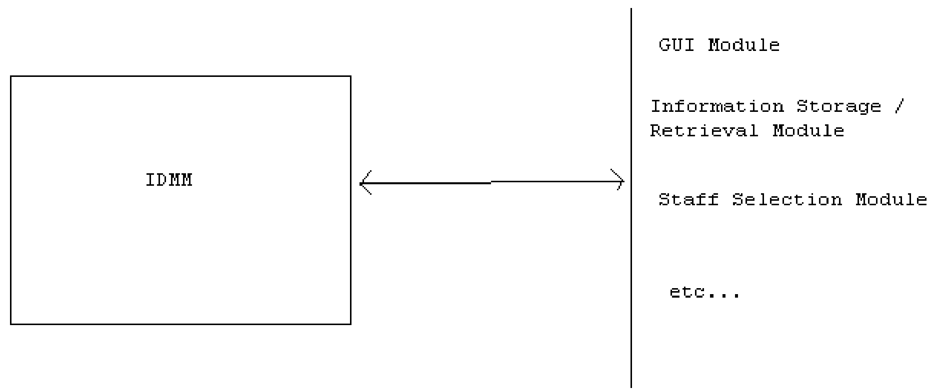
Total estimated time for code completion:  23 days given non-concurrent work.

Testing:  Tested via black box and sample testing.
Time to test module functionality:  one week.

Completed module:  One month (30 days) given non-concurrent work.

Input / Output specifications:

Access to data members of Manage 2000 for use by other modules will be performed through get / save function calls to the appropriate class data type.  Any information loaded or saved will pass through this module.  Interactions are diagrammed below:




## 2.3 User Interface Module (GUI)

Manage 2000 is designed to provide users a windowed system (GUI) that is split into three main areas. These areas are staff characteristics, project management, and skills tracking.

The first, staff characteristics, is the section of the interface devoted to information regarding staff information. Within this section, mangers will be able to view, print, update, add, and delete both information about the staff members and the actual database entry for the employee. The employees will be able to view, in report format, their current information, submit changes to a supervisor to be written to the database, and generate a resume.

The second area, project manager interface, serves as a control area for projects. Here, projects can be viewed, printed, updated, modified (based on changes made to the host program), added, and deleted. Project mangers may also replace individuals or reallocate individuals from on project to another within this area, as well as add new staff members to projects.

The third area, skill-tracking interface, allows a project manager the ability to view, add, remove, or modify any or all skills pertaining to the database. The information viewed about a skill is primarily a summary of the skill as well as its applications. These skills pertain to each qualification class preferences, such as education, technical schooling, current employment experience, previous employment experience, etc.

I) Functional Requirements

   A. What it will do

      i.) provide users with an efficient graphical user interface (GUI)

   B. What it will not do

      i.) access information from Project 98's pull down menuing system ii.) allow employee's to permanently modify their own (or any) database records

iii.) will not automatically assign recommended team members to projects

II) Staff Interface

A. Managers

i.) update, view, and send information to print a hard copy of staff characteristics

ii.) add, delete, or modify employee records

B. Employees

i.) view and send information to print a hard copy of their own characteristics

ii.) submit a modified record to supervisors for final posting to the database

III.) Project Management Interface

A. Project information

i.) view and submit print jobs

ii.) modify, add, or delete project records

B. Availability of individuals

i.) color-code individuals by level of availability

ii.) view the availability of all staff members

IV.) Skills Tracking Interface

A. View skill characteristics

B. Add, delete, or modify skill characteristics

V.) Input

A. Manager input

i.) employee information (name, social security number, salary history, etc.)

ii.) skill traits (c++, java, solaris, perl, etc.)

B. Employee input

i.) request changes of personal information

ii.) request changes of personal traits

VI.) Processing

A. Dependent on machine level hardware and number of requests within the database

VII.) Output

A. Console output and print requests for managers or employees

B. Color text for each employees availability

C. Change of information to the database records

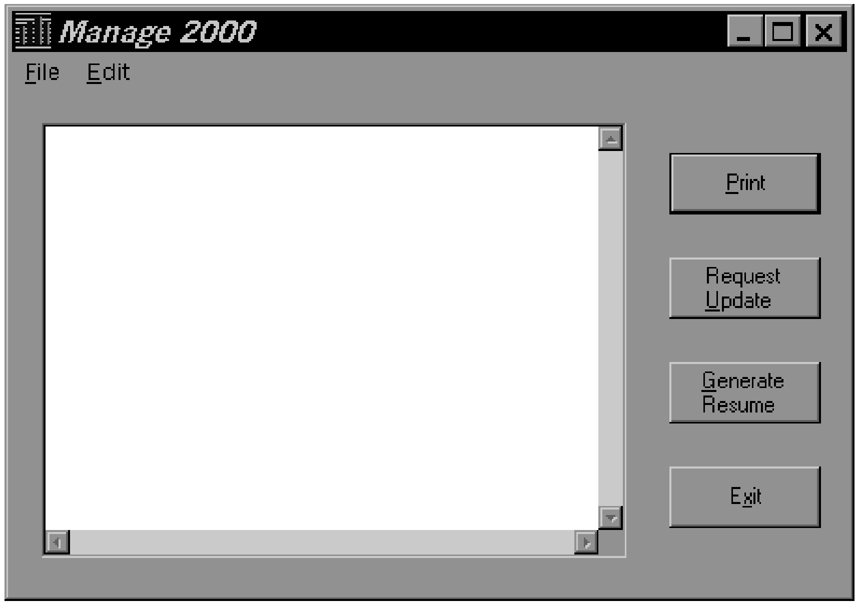
### 2.3.1 Scheduling

| Task Number | Description | Predecessors | Schedule |
|---|---|---|---|
| 1 | Come up with code | | 1 week |

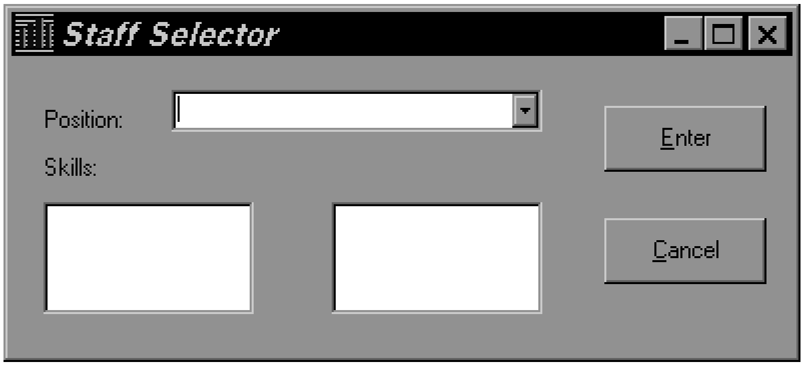
| | | | |
|---|---|---|---|
| | design | | |
| 2 | Come up with GUI | | 1.5 weeks |
| 3 | Integrate the code design and GUI | 1, 2 | 2 weeks |
| 4 | Test module | 3 | 1 week |
| MILESTONE | User Interface Module | | 5.5 weeks |

### 2.3.2 Manager Interface


### 2.3.3 User Interface


Manage 2000
File
Edit
Print
Request Update
Generate Resume
Exit

Staff Selection Module




## 2.4 Staff Selection Module Milestone

Subtasks:

- Come up with the algorithm, including flow chart
- Program functions for calculations and data retrieval
- Test correctness of functions
- Create GUI in Visual C++
- Program buttons with appropriate function calls

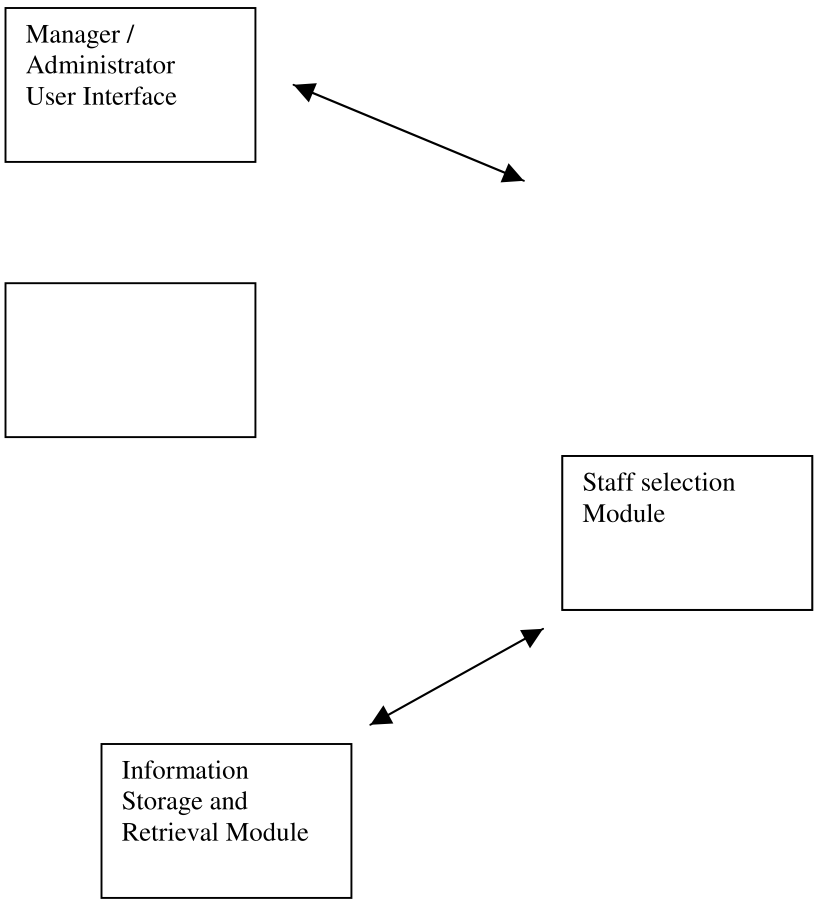
Milestone:

- Deliver finished module

Schedule:

- Algorithm: no more than one day
- Function programming and testing: 2 weeks
- GUI creation and programming: 2 days

- Total (with padding for sick days and such): 3 weeks



### 2.5 Report Generator Module Milestone

Subtasks:

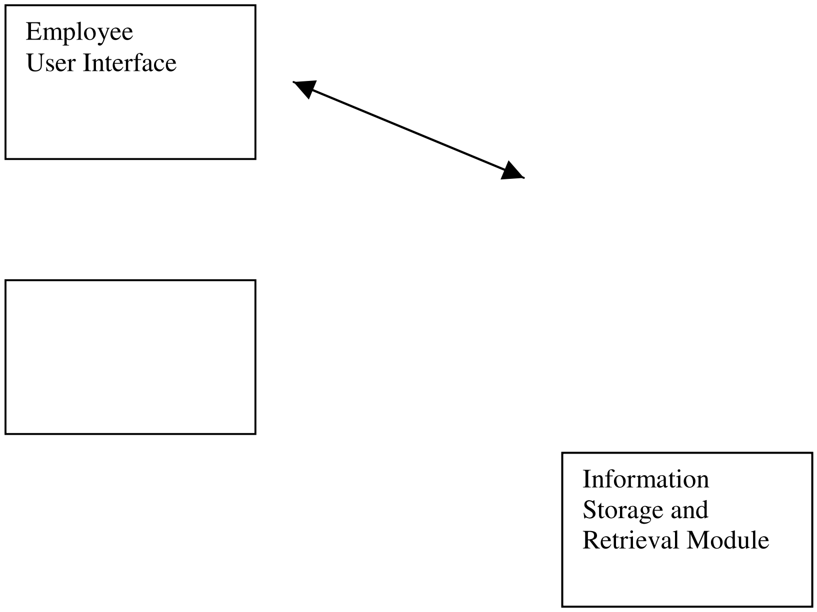
- Determine algorithm
- Program function
- Test function
- Add function call to the menu of employee view of his record (“Print” in File menu)

Milestone:

- Deliver finished program

Schedule:

- Algorithm: 1 day
- Programming and testing: 1 week
- GUI programming: 1 day
- Total (including padding): 2 weeks




**2.6 Modification Request Module Subtask:**

1. Describe abstract model detailing processes involved
2. Identify all classes & procedures to be reused
3. Identify procedures to be created w/in request module
4. Devise diagram depicting flow of data among procedures & modules
5. Develop and implement source code for request module
6. Develop and implement test cases for request module
7. Develop and implement user interface procedures for request module
8. Verify/Validate correctness of module implementation

Milestone: Request module tested code.

Schedule of subtasks:

Steps 1-4: 1 day

Step 5: 4 days

Step 6: 3 days

Step 7: 3 days

Step 8: 3 days

Schedule Allotment of days: 14 days (not including unexpected events)

**2.7 Modification Verification Module Subtask:**

1. Describe abstract model detailing processes involved
2. Identify all classes & procedures to be reused
3. Identify procedures to be created w/in verification module
4. Devise diagram depicting flow of data among procedures & modules
5. Develop and implement source code for verification module
6. Develop and implement test cases for verification module
7. Develop and implement user interface procedures for verification module
8. Validate correctness of module implementation

Milestone: Verification module tested code.

Schedule of subtasks:

Steps 1-4: 1 day

Step 5: 4 days

Step 6: 4 days

Step 7: 3 days

Step 8: 3 days

Schedule Allotment of days: 15 days (not including unexpected events)

Testing

One important way of testing code modules is 'black-box' testing in which the code is executed w/ sample test cases of input and output. After reading through the specifications, test cases will be drawn up to check whether the code obeys the specification document.

The test plans (revealed in future assignment) will specify:

- what testing is to be performed
- when it is to be performed
- and how it is to be performed.

## 2.8 Dynamic Refresh Module

Introduction

The Dynamic Refresh Module (DRM) is designed to automatically update employee's records while they are assigned to any particular project. When tasks are completed or milestones are met, the DRM notices these changes and updates the employee records accordingly. This module minimizes the work that employees and supervisors must do to ensure updated records. As the name implies, this is a dynamic function. Hence, there is no human interaction. The module periodically inspects the Project98 files and notes any newly completed tasks.

Goals

Recognize project task completion

Accurately update employee records

Save employees and supervisors maintenance time

Operational Scenarios

As employees are assigned to projects they will gain added experience or newly gained experience. When a project manager confirms that tasks have been completed, the DRM senses these changes upon its next inspection. This chain of events then call for changes to be made to the employee records. The records are updated and the DRM has successfully maintained up-to-date records.

Requirements

I) Functional Requirements

   a) What it will do

      i) Dynamically receive updated information about project status

      ii) Update personnel records regarding skills and traits gained as projects progress

   b) What it will not do

      i) Modify Project 98 data files

      ii) Delete current entries within employee records

      iii) Prompt employees or managers for current information

II) Input

   a) Project progression from Project 98 files or Manage 2000 database information

III) Processing

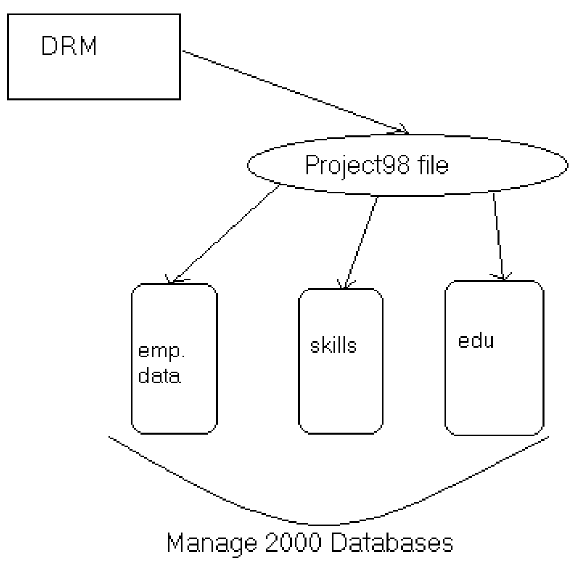
   a) Internally updates the employee records

IV) Output

a) Output is not immediate, but user records are displayed when requested by the client

Design

The DRM will interact directly with Project98 files. As the project files are modified, the DRM inspects the changes. It then calls the information storage module to store the modified data to the proper employee records. Since the project positions have previously been defined, there is no need for the DRM to add new tasks or skills to the database. Simply put, the DRM notes changes in the project file and writes updated information to the Manage 2000 database.



Milestones

Project98 file Query

Retrieve project98 file format Project98 Contractor 1 week

Implement queries Junior Programmer 2 weeks

Database update function

Open file call Junior Programmer 1 day

Write data to records Junior Programmer 3 days

Close file call Junior Programmer 1 day

Testing

Functionality can be confirmed by making changes to a project file where the database modifications are a known value. If the proper modifications are made, the module succeeds. On the other hand, if the expected results do not match, the module is not completed.

## 2.9 Information Storage / Retrieval Module

Introduction

The Information Storage / Retrieval Module (ISRM) is the interface between all other modules and the Data Management Module (DMM). When other modules need to read or write to the database, they call the ISRM to help complete the task.

Goals

Retrieve requested data

Store data requests

Assist modules with database interfacing

Operational Scenarios

When a user makes a request to update his/her records, the data is written to a temp file. Then the supervisor either approves or denies the change. If the change is approved, the request is made to the ISRM which receives the data from the other module and writes it to the database. Also, if users wish to view their information, then the call is made to the ISRM and it retrieves the data and outputs it to the desired medium.

Requirements

I) Functional Requirements

A. What it will do

i. Retrieve data from within Microsoft Project 98 files

ii. Process information from various self inclusive files

B. What it will not do

i. Create, revise, update or delete Microsoft Project 98 data files

I) Input

A. Files /fields from Microsoft Project 98

B. Files related to internal procedures within Manage2000

C. Files containing stored information processed by Manage 2000

II) Processing

A. Files from Microsoft Project will be retrieved and various data fields among these files will be processed and potentially stored within a Manage 2000 database file.

III) Output

A. Display stored data to the device console

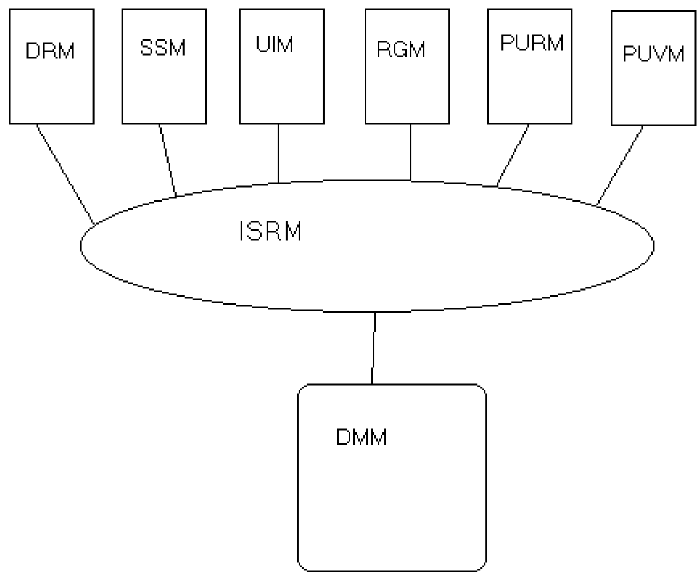
B. Generate printed reports (resume)

IV) Performance Requirements

a) Speed of storage and retrieval processing relies on database file size, number of files to process, host hardware, and other active database operations.

Design

This basically inherent to the database software that we have chosen.  Custom functions must be written for this module to interface with the other modules.




Milestones

Read function

Read data from the database        Junior Programmer

3 days

Write function

Write data to the database        Junior Programmer        1 week

Testing

This module must be tested with each module as well as by itself. Known results testing is easiest with this module. As you know what the result change should be, you can determine if the read or write request to the database is correct.