

# **Table of Contents**

<b>1. Introduction</b>	2
a. Initial Situation and Goals	2
b. Notations Used	2
c. Conceptual Model	3
<b>2. System Operation and Environment</b>	5
<b>3. Functional Requirements</b>	6
<b>4. Database Requirements</b>	7
<b>5. Nonfunctional Requirements</b>	10
a. Security	10
b. Reliability	10
c. Portability	10
d. Extendibility	10
10	
e. Response Time	10
<b>6. User Interface Requirements</b>	11
<b>7. Error Behavior</b>	13
<b>8. Documentation Requirements</b>	14
<b>9. Acceptance Criteria</b>	15
<b>10. Maintenance Requirements</b>	16
<b>Glossary</b>	17
<b>Index</b>	19

## 1. Introduction

### a) Initial Situation and Goals

If you are an university student in today's world, one of the most tiring and annoying duty in your life may be the registration process, that is what you have got to do in the beginning of every semester. This process is rather a complicated and a long one which most students hate to be part of. There is no doubt that the registration is a must for both the students and the university registrar. From the students point of view, the choice decision of the courses that will be taken in next semester, becomes really easy with an every-semester-registration process. The student may no more wander around to choose courses and find himself/herself in panic. Also, the registrar is sure that every student will register and take the courses as planned.

The interesting point about registration process is that although, it is a time consuming process, it is also a routine one. At this point, the idea of a software doing the registration for every student appears as a good idea. This software may take one person in front of the screen at a time and do his registration, update records, inform registrar and give various kinds of help to the student. The strong points of the idea are that it decreases the lost time considerably, makes registration easy for both the student and the registrar and correct the errors.

In this software product which is named *ARU* (Automated Registration Utility), the described self-registration process is implemented. As the product development team, we hope that *ARU* will make life easy for students.

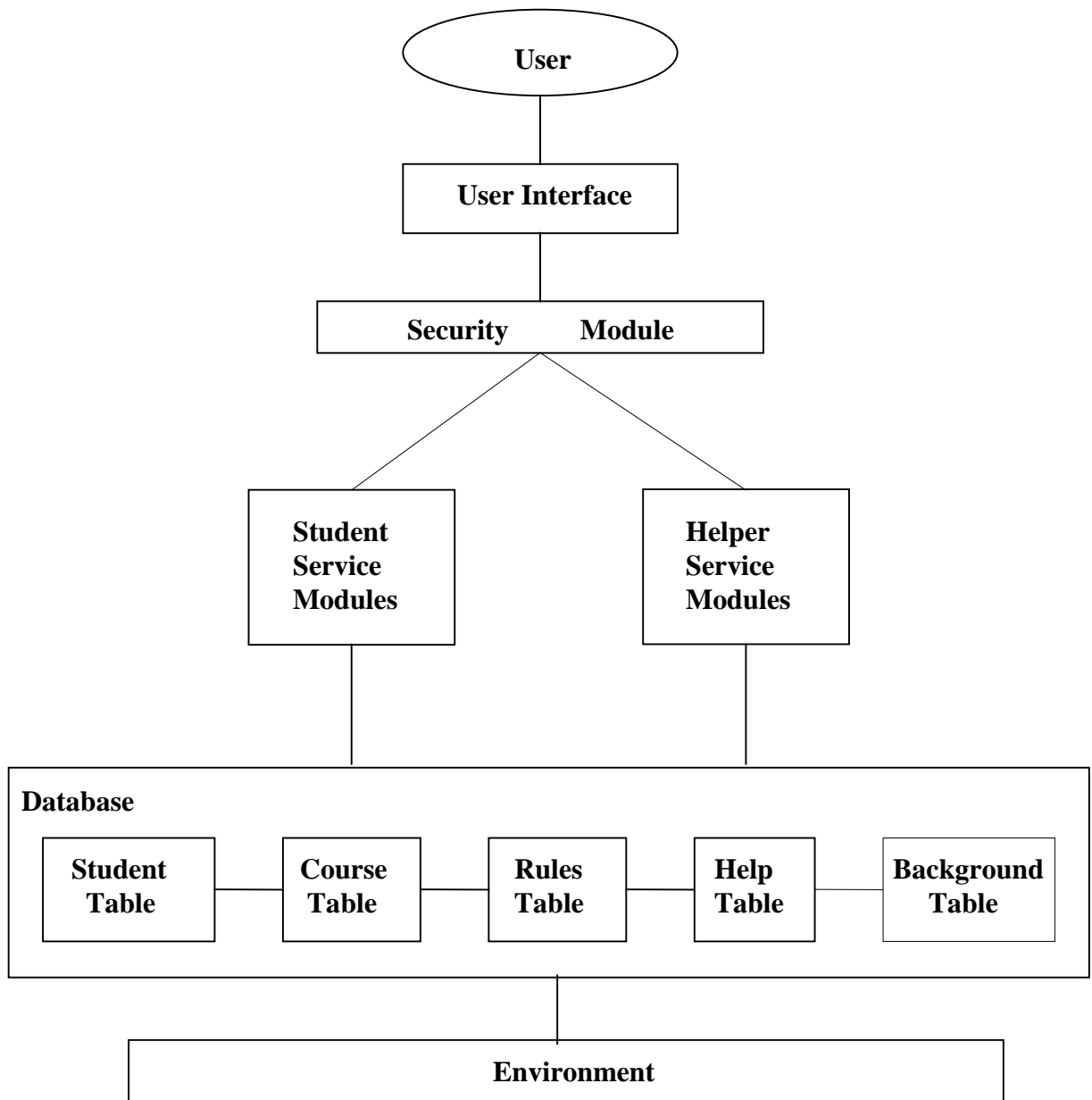
### b) Notations Used

Following notations will be used in documents from now on :

<b>bold</b>	: Used to show record and field names of the database. Also the lower headers are shown either <b>bold</b> or <u>underline</u> .
<i>Italic</i>	: Used to show special names like <i>IBM</i> , <i>Microsoft</i> , <i>Windows</i> etc...
<u>underline</u>	: Used to show the service names provided for the user.

**c) Conceptual Model**

Conceptual model of the software product is given below as a diagram :



In very bottom of the conceptual model of our software product, resides the environment on which the program will run and the databases are to be organized. The second level that is above the environment, is the main database. Here, there are four tables containing all data which the *ARU* will use to continue the registration process.

Above the database, the process modules exist. There are three main modules, as you can see from the diagram. But, as a conceptual design, security module is above the other two. In other words, the user will have to pass the security checks in security module, if he/she wants to access to the other two modules and the database. By-passing the security module, in any case, is strictly forbidden and extensively controlled. The last thing before the user is, of course, the user interface. This interface is to be as friendly as possible. And the real user concludes the model.

## 2. System Operation and Environment

Our software product, *ARU*, runs basically on a PC. But, it is a *Windows* environment program, so that particular PC has to satisfy the requirements of *Windows 3.1* in the first place. These requirements can be specified as follows :

CPU	: has to be <i>Intel 80286</i> up to <i>Pentium</i> .
Memory	: has to be 2 MB minimum.
<i>Windows</i> Version	: has to be 3.1 up to 4.0 (i.e. <i>Windows 95</i> )
Disk Space	: nearly 20 MB of hard disk space is needed.

### 3. Functional Requirements

This section introduces and fully explains the user services provided by the software product named *ARU*. Those services mainly stays above the database and control it. These functional requirements that our product will include can be summarized as follows :

- Student Service Modules
- Helper Service Modules
- Security Modules

**Student Service Modules** : This service is just for the students registering themselves in the student mode. Here, the user can do many specific operations about himself/herself. These includes ; taking advice from the advisor, looking back to his previous grades, getting the stamps from various checkpoints etc.

**Helper Service Modules** : As the name implies, this part is to be implemented just as the hypertext help system of *Windows* operating system. This system will supply on-line information through the registration process. With the help of this mechanism, the students can always find their ways on the registration. Hypertext system is an additional tool, organized to give more information on some special words or topics without losing the power of being on-line.

**Security Modules** : In addition to service modules, *ARU* contains a security-check module which is implemented for protection of the program and the students using it. This service accepts all new users as students and they are given unique account numbers. Every student enters a password and uses it to enter the system in future. Also, the students can save and return later to their registration processes until they do the last part of the registration (i.e. submitting the optic form).

## 4. Database Requirements

Automated Registration Utility is a fully database-grounded program. It uses a huge database consisting of five main tables and read/write data from/onto these tables. The tables, very briefly, are Student Table, Course Table, Rule Table, Help Table and Background Table.

The database containing these tables is created and fully prepared beforehand by using a Database Manager. During registration process students update it continuously. The structures of the tables are as follows :

Help Table : 

<b>Help ID</b>	<b>Help String</b>
----------------	--------------------

Rule Table : 

<b>Rule ID</b>	<b>Rule String</b>
----------------	--------------------

Course Table : 

<b>Course ID</b>	<b>Course Name</b>	<b>Instructor</b>	<b>Credit</b>	
<b>Opcode</b>	<b>DepCode</b>	<b>Pre 1</b>	<b>Pre 2</b>	<b>Pre3</b>

Student Table : 

<b>Student Name</b>	<b>Student ID</b>	<b>Class</b>	<b>Department</b>
<b>Semester</b>			

Background Table : 

<b>Student ID</b>	<b>Course ID</b>	<b>Grade</b>
<b>Status</b>		

In detail, these tables consist of the following :

**Help Table** : This table contains all the help strings and their associated help ID numbers. Although, it is shown as a part of the main database, this table will be separately implemented as a database in the first place. Help table is nothing more than a database for hypertext help system.

Fields :

- **Help ID** : Unique ID number for a particular help topic. <integer>
- **Help String** : Main help information. <string 20>

**Rule Table** : Various rules concerning the registration process is kept in this table. This has two fields ; an ID field and a string field containing the actual rule.

Fields :

- Rule ID : Unique ID number for a particular rule. <integer>
- Rule String : Main rule information. <string 20>

**Course Table** : This table is the actual representation of the main course schedule of the semester in the database. Its records contains exactly the fields that are common to schedules.

Fields :

- Course ID : Unique ID number for a particular course. <integer>
- Course Name : Name of the course in full. <string 40>
- Instructor : Name of the instructor. <string 20>
- Credit : Credit value of the course. <integer>
- OpCode : Optic form code of the course. <integer>
- DepCode : Department code. <string 4>
- Pre1 : First of the prerequisites in ID format. <integer>
- Pre2 : Second of the prerequisites. <integer>
- Pre3 : Third of the prerequisites. <integer>

**Student Table** : All the information about particular students are kept in student table. It contains not only student name, school ID numbers, class but also, semester number and department name.

Fields :

- Student ID : Unique ID number for a particular student. <integer>
- Student Name : Name of the student. <string 20>
- Class : Class of the student. <integer>
- Department : Department name of the student. <string 4>
- Semester : Number of semester of the student. <integer>

**Background Table** : This table is more like a history of the university students. In this table, all the grades of the courses that were taken by each of the students is kept separately. Background table is the main source of information and is very important for solving the problem of prerequisites.

Fields :

- **Student ID** : Unique ID number for a particular student.  
<integer>
- **Course ID** : Unique ID number for a particular course.  
<integer>
- **Grade** : The grade taken by the student in that course.  
<string 2>
- **Status** : This shows the course status, like NR for No Credit etc... <string 2>

Main database is to be created with *MS Access* and accessed by the help of ODBC. It obeys standard database rules and can be manipulated with standard database managers.

A simple calculation on the tables gives the approximate disk space necessary to organize such a database. It goes as follows :

- Help Table** >  $300 * 22 = 6600$  bytes for 300 help records.
- Rule Table** >  $300 * 22 = 6600$  bytes for 300 rule records.
- Course Table** >  $500 * 76 = 38000$  bytes for 500 course records.
- Student Table** >  $20000 * 32 = 640000$  bytes for 20000 students.
- Background Table** >  $1000000 * 10 = 10000000$  bytes for 1000000 background records.

**Total** :  $10691200 + \text{Additional Buffer (approx. 4 MB)} = 15 \text{ MB}$

So, for a full database of the explained kind at least 15 MB of disk space is necessary.

## 5. Nonfunctional Requirements

**a) Security** : Security of the data in database, is maintained in two ways. First, the database can only be controlled by standard database managers and *ARU* uses ODBC to access the data. This guarantees the integrity of the database. As the second point; users are allowed to reach data only if they have a valid account. And according to their account types, several restrictions are provided. This disallows the unauthorized users to destroy the data.

**b) Reliability** : All the tool operations work perfectly. No matter how large is the database, it is maintained and protected efficiently. *ARU* never breaks down.

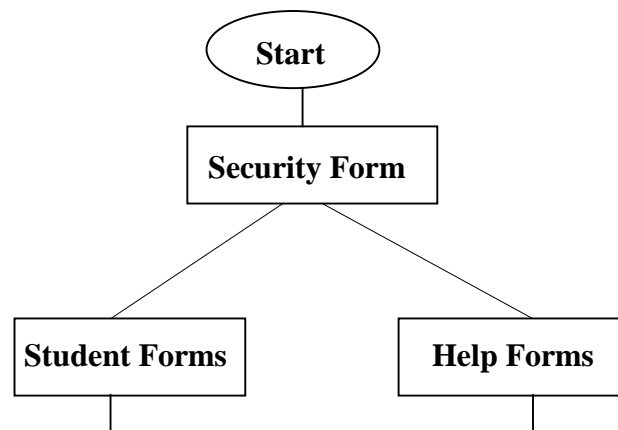
**c) Portability** : The tool used to develop this software product is *Borland's Delphi*. It generates .EXE code that is compatible with and executable on any 80286 or higher processor based PC. Of course, it is a *Windows* environment utility.

**d) Extensibility** : The program, with the help of the *Delphi's* IDE, is very modular indeed. So, when needed, additions, updates and alterations can be made easily on the source and forms. Nothing is too glued to each other.

**e) Response Time** : The program has two parts ; modules and database. The modules are all to be implemented as basic as possible and they will not contain any algorithm that has exponential space or time complexity. Also, the database access is maintained by ODBC, one of the fastest methods for *Windows* environment.

## 6. User Interface Requirements :

Our software product works on many forms that are related to each other by various modules. Mainly there are three form collections each having many forms inside. As a diagram, this can be shown as the following :



**Security Form** : This form is a simple one which uses and updates personal records of students. It consists of input fields to take first information about the users and OK-EXIT buttons. Security form has no menu inside. According to the validation of the user, he/she is rejected or accepted to the system.

**Student Forms** : Student Forms provide the services that are necessary for students in the registration process. There is one form used to get information about the schedule. It consists of input fields to demand information and output fields which contain the wanted information. Also the student's transcript can be examined on a transcript form which is very user-friendly with its list box, scrolling bars etc... This form collection, generally, work on pull-down menus and control buttons to manage operations.

**Help Forms** : Help form collection organizes the help texts of the software product. The structure of forms used here, is very like the *Windows Help Manager*. Hypertext system uses the **Help Table** in the database to provide on-line information about some words in the help forms. These information comes to the screen as a pop-up window. The special words that contain hypertext help, is shown in different color in the help forms.

All of the services of the program are implemented under pull-down menus and control button. In addition to these, *Windows* environment makes the user interface incredibly friendly. Menu option names are, most of the time, self-explanatory, but as the product development team, we will give detailed explanation about the usage of the program in user manual.

## 7. Error Behavior

Automated Registration Utility (*ARU*) is to be designed and fully tested against different stress test conditions. All the errors found in these test are to be all corrected, so run time errors are irrelevant to our software product.

This program runs only under *Windows* environment. If user tries to run this program under other environments, it will not work.

If program tries to open a non-existent file, errors will be occurred. Loss of any related file to this program creates errors.

In any error case which breaks the *Windows* environment down, the last state of the program is conserved. Also, the database is kept in the last update state. Our product is reliable at the limit of *Windows*' reliability.

User errors are not possible in *ARU*, because all the services and functions are menu driven. Moreover, every user action is checked against the registration rules. But when user wanted to print transcript and no printer was connected to the computer or printer was not ready to service, user will be warned.

## 8. Documentation Requirements

Documentation will be as basic as possible. And they will be ready on time. According to your wish, we can submit you documentation on paper or on electronic media. Schedules for the documentation is arranged as follows :

Requirement Definition Document (draft)	Oct.	16, 1995
Requirement Definition Document (revised)	Oct.	23, 1995
Preliminary User Manual (draft)	Oct.	30, 1995
Preliminary Design Document (draft)	Nov.	13, 1995
Detailed Design Document	Dec.	28, 1995
User Manual	Dec.	28, 1995

## 9. Acceptance Criteria

Acceptance Criteria, for each of the individual requirement of the software product is given below :

**Documentation** : Documents will be in *Word* format, reside on paper or electronic media and be as basic as possible. All will be delivered on time.

**Database** : All tables and records of database will be compatible with standard database managers. In short, all the database will be controllable by *Access* or like products.

**Program** : Every module will work properly and accordingly to user requirements. No errors have to be encountered.

**Alpha Test** : Program will be tested for the first time. The errors occurred during the Alpha Test, are to be corrected until the Beta Test by our installation and test team.

Positive result of the Beta Test will imply that all the requirements are met.

## **10. Maintenance Requirements**

Our software company declares that the installation of the program will be made by our installation and test team. Every possible problem during installation will be corrected by this team. The manuals covering all the product, will be provided on time. Also, when new versions come around, you will be informed about them and offered special chances to get them.

Our company will arrange seminars to give information to your people about this product as soon as possible. This seminars will last for 3 or 4 weeks. Any other information request will be granted immediately.

As a last point, we provide you the program with full database organized according to your will.

## Glossary

<b>ARU</b>	: Automated Registration Utility.
<b>CPU</b>	: Central Processor Unit.
<b>Database</b>	: Consist of some collection of persistent data that is used by the application systems of some given enterprise.
<b>Delphi</b>	: Visual development environment produced by Borland.
<b>Disk space</b>	: Space required for execution of ARU.
<b>Form</b>	: Visual component of the program.
<b>Hypertext</b>	: Highlighted text to supply help.
<b>IBM</b>	: International Business Machines.
<b>IDE</b>	: Integrated Development Environment.
<b>Module</b>	: is a manageable component of a single software.
<b>MS</b>	: Microsoft.
<b>MS Access</b>	: A database management system produced by Microsoft.
<b>PC</b>	: Personal Computer .
<b>Pentium</b>	: Microprocessor designed by Intel Co.
<b>Processor</b>	: A semi-conductor device (integrated circuit) that is manufactured by using large scale integration technology. It includes arithmetic & logic unit, register arrays, and control circuit on a single chip.
<b>Record</b>	: Data storage unit which consists of some fields.
<b>Response Time</b>	: This is the amount of the time it takes to start responding .

<b>Source</b>	: Text component of the program that is understandable by compiler.
<b>Windows</b>	: Visual operating system developed by MS.
<b>Windows 95</b>	: First 32 bits version of MS Windows.
<b>Word</b>	: A word processor designed by MS.
<b>.EXE code</b>	: Executable code of a program.

## Index

### A

ARU 2, 3, 5, 6, 10, 13, 17

### B

bold 2

### C

course table 3, 7, 8, 9

CPU 5, 17

### D

database 2, 3, 4, 6, 7, 8, 9,  
10, 11, 13, 15, 16,  
17

database manager 7, 9, 10,  
15, 17

Delphi 10, 17

disk space 5, 9, 17

### E

extendibility 10

### H

help table 7, 9, 11

hypertext 6, 7, 11, 17

### I

IBM 2, 17

IDE 10, 17

italic 2

### M

memory 5

Microsoft 2, 17

module 3, 4, 6, 15, 17

MS Access 9, 17

### O

ODBC 9, 10

### P

portability 10

processor 10, 17, 18

PC 5, 10, 17

### R

reliability 10, 13

response time 10, 18

rule table 7, 8, 9

### S

security 3, 4, 6, 10, 11

security form 11

security module 4, 6

student form 11

student service modules 6

student table 7, 8, 9

### U

underline 2

### W

Windows 2, 5, 6, 10, 11, 12,  
13, 18

© 1995 by Totem Software Development Company

All rights reserved. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording and/or otherwise, without the prior written permission of the Totem Software Development Company.

*ARU* is a registered trademark of Totem Software Development Company.

*IBM* is a registered trademark of International Business Machines Corporation.

*Windows, MS Access* are registered trademarks of *Microsoft* Corporation.

*Pentium* is a registered trademark of *Intel* Corporation.

*Delphi* is a registered trademark of *Borland* Corporation.