QuickAdvise - The Search for a More Efficient Method of Advising

Gregory G. Pengiel '94

Recommended Citation
http://digitalcommons.iwu.edu/cs_honproj/18
RESEARCH HONORS RECORD

Name: Gregory G. Penciel

Research Field: Computer Science

Date of declared intent: 9/9/93

GPA (cum.) on above date: 3.94

GPA (in field) on above date: 4.0

Number of credits earned (total) by above date: 28

Number of credits earned in field by above date: 8

Hearing Committee Members:

- Project Advisor (Convenor): Dr. Lisa Brown
- Member from Outside Field: Dr. He
- Others (2):
  - Dr. Susan Anderson-Freed
  - Dr. Robin Sanders

Student Approval: [Signature]

Faculty Consent: [Signature]

Please attach project proposal

Candidates DO NOT WRITE BELOW THIS LINE

Project Hearing Date: 5/13/94

Committee Decision:

Designate Research Honors: [Signature] Date: 5/13/94

Designate No Honors: [Signature] Date: 5/13/94

Signed (Project Advisor): [Signature]

Signed (Associate Dean of Academic Affairs): [Signature] Date: 5/13/94

A final copy of the Research Honors papers must be returned with this form before the Research Honors designation is granted.
QuickAdvise - The Search For a More Efficient Method of Advising

In order to choose courses both efficiently and properly, a student and advisor must look at which courses have been taken, and also which ones are necessary. They must then determine whether or not the student is qualified to take the necessary courses. Unfortunately, there is no easy way to record and maintain this work so that it may be used throughout the college career. Hence, even though the student and advisor recently determined which classes were needed, they must once again look up all relevant information and determine which courses are best. This is exactly the type of problem that a computer can solve. However, in order to properly design a software application, various aspects of the situation must be extensively researched. First, the solution presents a quandary in terms of the system design. The language that is best suited to this type of application must be determined. Secondly, in order to be useful, the user interface must be well constructed. People do not like software that is not user friendly, so to make this a worthwhile software application, the interface must be both intuitive and accommodating. Finally, different development platforms must be considered in order to properly solve the problem. If the previously mentioned items are properly devised and documented, the actual construction of the software should not be difficult. Other points to consider are testing documentation, unit and system testing, code reviews, scheduling, and graphic user interface design.