

Design of Online Student Fee Committee  
Proposal Information System

By  
Kenneth Cairns  
Greg Jacobson

A Senior Project submitted  
in partial fulfillment  
of the requirements for the degree of  
Bachelor of Science in Industrial Engineering

California Polytechnic State University  
San Luis Obispo

Graded by: \_\_\_\_\_ Date of Submission: \_\_\_\_\_  
Checked by: \_\_\_\_\_ Approved by: \_\_\_\_\_

## Abstract

The purpose of this project is to overhaul the IME department's Student Fee Committee (SFC) proposal system. A web enabled database was designed in order to improve this system. A significant amount of time was spent researching different programming languages, paradigms, and frameworks in order to determine the most logical approach to the system redesign. After a few false leads, it was determined that the most appropriate solution to the problem was to design a C# based web form application using asp.NET technology, developed in Microsoft Visual Studio. A mockup was created in Access in an attempt to determine if the table design would be suitable and deliver the required customer needs. The Access model straightened a few kinks out in the initial design, and Visual Studio was used to build the interface and Access interconnects. Currently the web interface is still moving through the final development process. Some aspects such as secure login are currently in the works, but the basic I/O between the frontend and the backend is complete.

## Table of Contents

Abstract.....	2
Table of Contents.....	3
Table of Figures.....	4
Introduction .....	5
Background .....	5
Literature Review.....	7
Human Factors.....	7
Database Design.....	8
Project Management in Website Design .....	9
Design.....	12
Web Interface .....	12
Defining User Requirements.....	12
Determining Platform .....	12
Compiled Languages .....	12
Interpreted Languages.....	13
Investigated Programming Frameworks.....	14
The MVC framework .....	15
Human Factors Considerations.....	17
Platform Constraints .....	17
Database .....	17
Defining Requirements .....	17
Economic Analysis.....	18
Methods.....	19
Web Interface Design.....	19
Database Design.....	20
Proposals.....	20
Users .....	21
Report .....	22
Economic Analysis.....	27
Conclusion.....	27

Works Cited.....	28
Appendix A (Figures).....	30

**Table of Figures**

Figure 1: Model View Paradigm.....	16
Figure 2: Current SFC Proposal Process .....	18
Figure 3: Proposed SFC Proposal Process.....	19
Figure 4: Proposals Table Field Data.....	20
Figure 5: Item List Table Field Data.....	21
Figure 6: Proposals and Item List Relationship Structure.....	21
Figure 7: Members and Login Relationship Structure .....	22
Figure 8: Proposals Report Printout.....	23
Figure 9: Previous Proposals Submission Template .....	24
Figure 10: Proposed Proposals Submission Template.....	25
Figure 11: Proposed View Consolidated Proposals Page.....	26

## **Introduction**

Our project is to overhaul the IME department's Student Fee Committee (SFC) proposal system. To do this we have designed a database which will handle all aspects of the proposal process, enabling the SFC administrator to spend his or her time on more important aspects of their job.

## **Background**

In 2002, an academic fee was voted in and approved by Cal Poly students. The fee was introduced in order to improve the quality of academic programs, whether it be through additional course offerings, lab upgrades, or department related projects and programs. The IME department has formed the Student Fee Committee which plans and oversees the allocation of the funding.

Every quarter, the SFC chair sends out the application for funding to the IME student body via email. All of the completed funding requests are then sent back, and printed out before the meeting. The applicant must attend the meeting, and make his or her case to the committee, where a ruling is made as to the appropriateness of the proposal. Proposals must be tracked after a ruling is made as well. Although this is an effective way to manage proposals, it's also very time consuming, and we feel that a dedicated proposal submission and management section on the IME web site will save money and time in the long run.

We were recently contacted by Joe Anderson, current SFC chair, regarding producing this dedicated proposal interface and making it available on the IME web site.

At the core of the web page would be a database, with a form designed for students and or faculty to submit their proposal. An additional set of tools will be required to allow Stephanie

and the SFC chair to access all of the proposals, and edit them. Putting this into action requires database design skills, project management aspects, an economic analysis, as well as a firm grasp of human factors engineering to streamline the user experience.

## Literature Review

### Human Factors

Besides the database backend functioning, the most important part of our project will be a good user interface. Simplifying the input of data into the system will make the system more straightforward for the person making the proposal. An interface to view and add comments and criterion for each proposal will also be crucial for both Stephanie and the SFC chair. Both of these aspects will streamline the process, while simultaneously freeing up time for Stephanie to attend to other business.

The economic benefits of this project would stem entirely from time savings. In order to maximize the time saved, we will structure the site in a familiar and effective way. Information will be grouped based upon conventional practices commonly known by the user. Identifying information regarding the fee request, the proposers name, amount requested etc. will be grouped together. We will try to keep information delineated in a way similar to other applications. Creating a familiar user experience will be far more effective than trying to reinvent the wheel.

The ultimate layout of information on the proposal submittal form will have a large impact on how the end user actually interfaces with the system. Since the SFC proposal has so much information required to process it, we want to provide the most straightforward interface to the user. A balance must be struck between providing the most program functionality and the simplest user interface, as there is clearly a tradeoff between these aspects.

## Database Design

One of the most important aspects in database design is interacting with the customer or client (Yang 2010). What is most important with this web tool is to provide easier access to student fee proposals for all advisors involved (Allen 2010). In order to do this, information must be accurately and effectively stored. With an approval process involving multiple authorized advisors, the system must incorporate a login process. This has been implemented previously at Cal Poly in card-reader form using the student database (Mehas 2010). The Student Fee Committee will require a more advanced online login process, where new members can be easily given access to the restricted information. This kind of technology has only been described theoretically and will require further meeting with the network administrator for the IME website (Allen 2010). Throughout the process of designing and implementing this database, there will be many meetings and revisions with the client; the Student Fee Committee.

In order to get a clear understanding of similar implementations in this area, other student fee proposal systems around campus were analyzed. The most similar design of proposal systems was with the Cal Poly Mechanical Engineering Department's student fee committee website (MESFAC 2010). One of the first notable issues with this website is the lack of concern for human factors. The website is unfriendly and cluttered and has only basic database functionality. While the scope of this project is only to build a module for the current IME website, it is important to note the areas in which improvements can be made. Human computer interaction is an important part of interacting with the database and simplicity will go a long way in both the graphical user interface and logical model of the database design. One

of the issues outlined by Stephanie Allen was the way in which the proposals were submitted and displayed (Allen 2010). Research will need to be done in order to determine the best way to both submit and show proposals, and also track information in the database.

The most important technical issue of the database implementation is the restrictions of the website equipment available (Allen 2010). As outlined in Nick Mehas' senior project on Facility and Database design, one of the most important obstacles in an online database implementation is the possibility for inconsistency of data (Mehas 2010). This is also addressed as a common issue when testing and troubleshooting .NET on the host server versus a personal computer (Yang 2010). Working with the network administrator should alleviate this problem.

Quality control is noted as one of the primary components of building a proper database (Sun, Lu 2009). This will be done with a strong logical model of the database as well as a simple and efficient interface (Yang 2010). Most references on database design are theoretical and much more advanced than is relevant for this topic. This scale of this database design is similar to that of a small business and should be treated as such.

Through the production of our software, it will be necessary to test and debug code to validate the database structure, as well as ensure consistency between server and client side. It is important to note that the final iteration of our software will be interfaced with solely over the internet, so there cannot be any errors in the online interface.

### **Project Management in Website Design**

Project management in software or website design is a difficult task. Coding can be error prone and task durations have a large variability. In order to reduce software product development time, it's necessary to maintain many tasks concurrently (Callahan, Moretton

2000). In our case, that means getting set up physically with the website location at the same time we're coding out the information system. In project managing the development of a website, it is important to manage user expectations because of the tendencies to have scope creep (Petter 2008). The effectiveness of project management techniques depend heavily on the accuracy of the task duration estimates. In software development, a test was conducted and found that while the majority of tasks were overestimated, the mean error is about a 1% underestimate (Thomas, Allen 2000). The recommended technique to use in order to deal with this is to further break down each task in your work breakdown structure to provide more accurate task duration estimates.

The role of project management in the implementation of information systems used to entail simply the design and implementation of the project. The future of project management for information systems has now moved toward the entire product life cycle including maintenance and phasing out (Ahlemann 2008). The model which would be most successful for the student fee committee would be a system that allowed for easy maintenance and transfer of knowledge over the years. There will be many people maintaining this system over the years and it should require little to no modification in order to incorporate future needs.

Project management in software development and website design is an emerging field. Currently it consists of project managers spending their time trying to encourage the programmers and the real time estimate data comes directly from the programmer. The programmer may or may not have the experience necessary to accurately determine task duration estimates. Using the steps in order to estimate the project lifecycle, a rough outline can be created, but when it comes to programming, the majority of the work will be in

controlling the dates during coding. An appropriate risk management analysis must be evaluated due using a risk severity matrix with respect to impact on the critical path of the project lifecycle (Clifford, Erik 2003). Because of the inherent variability with coding, it will be very important to keep close management on the entire coding part of the project. Variability of real task duration times from the previous task duration estimates will help in revising future task duration estimates (Clifford, Erik 2003).

Due to the nature of this project, crashing of tasks is not a realistic solution. The senior project guidelines require a set number of hours dedicated toward the project and after exceeding that, no major features will be added unless economically justified. Using a model to test the trade-offs for time, cost and quality, important decisions can be made as to what can be included in the website (Khang, Myint 1999). In most cases, priority will be given to the features which will save time for end users in the future or are economically justified otherwise. According to (Huq 2000), adding in testing tasks earlier on in the project life-cycle will prevent the chance of complications later on in the project. Testing is one of the most important parts of a successful information system implementation and this task cannot be compromised. This will also reduce future maintenance costs and technical issues, which could arise much later.

Because of the tendency for scope creep on website designs, a clear project scope must be used to maintain focus. This entails maintaining the core features and functionality required to begin with. Further on in the process life-cycle, the availability of extra capacity can be reevaluated (Allen 2010). In this case, the quality of the deliverable which is turned in will have a direct impact on the reputation of the IME department. In managing this project, the constraint will be quality primarily.

## **Design**

### **Web Interface**

The interface determines the only quantifiable cost and time savings for the administrator. As such, it's the most crucial end result of the application. The human factors considerations of the forms will be covered in a later section.

### **Defining User Requirements**

The main requirement of the customer was an overhaul of the SFC proposal system. The information contained in the proposal stayed the same; the delivery of this information was the only thing that changed. The contents of the proposal have stayed the same.

### **Determining Platform**

At the onset of this project we determined that a great deal of consideration should be placed on selecting the appropriate tool for the job. We had some essential programming skills under our belt through our Visual Basic background, but we wanted to do some research to see if branching out would be beneficial. As such, we brainstormed some possible solutions, and selected 4 programming languages for further research. Additionally, we found that for many of these programming languages there were associated frameworks to enhance the functionality. Here is a brief summary of our findings

### **Compiled Languages**

#### ***Visual Basic***

Visual Basic was an appealing option since we have a solid background in it through our work in IME 312. The structure of the language is very simple, which makes it easy to get

started. Additionally, VB has an integrated interactive development environment (IDE), which allows developers to generate code through selections in menus and forms. The IDE facilitates simple connections between a GUI frontend and the logic functions tied to the GUI elements. An event driven language allows great flexibility to the developer, but doesn't ensure that functionality is necessarily pragmatic. Additionally, VB offers object oriented features, but is not a fully object oriented language.

## *C#*

C# was another appealing choice. We are familiar operating within the Visual Studio environment. C# (along with VB) is one of the programming languages which are compliant with the Common Language Infrastructure (CLI) specification developed by Microsoft. Other features of C# include that it is object oriented and strongly typed. Object orientation is the main feature which we have deemed necessary for our project. Other features will have an effect on how our application is structured, but will not make a critical difference like object orientation will.

## **Interpreted Languages**

### *Python*

Python is a dynamic, interpreted, object oriented programming language. Python is useful for a wide range of applications, including scripting purposes. Python has a strong emphasis on readability, which makes it approachable, while retaining high-level functionality. An interpreted language attempts to execute every line of code at the last second. When the interpreter discovers errors, an exception is raised, and the stack trace is printed out.

## *Ruby*

Ruby is a dynamic, open source language with a strong focus on productivity. It's creator, Matz, wanted to create a scripting language that was more powerful than Perl, and more object-oriented than Python. Ruby is the only programming language where everything is an object. Other languages have building blocks which cannot be operated on by methods which are called primitives. The number 5 for example is considered a primitive in all other languages. In Ruby, a method can be called on 5 because it's an object. One of the most basic examples of this functionality would be:

```
5.times {print "We love Ruby!"}
```

Which would output:

```
We love Ruby! We love Ruby! We love Ruby! We love Ruby! We love Ruby!
```

Ruby was also a very appealing choice based on the web framework, Ruby on Rails. The advantages of Ruby on Rails will be summarized in a following section.

## *Investigated Programming Frameworks*

Every framework we researched can follow a newly popularized Model View Controller paradigm. This model follows a "Don't Repeat Yourself" methodology. Every bit of programming logic has an appropriate place in the framework, and needs to be placed in the proper location for ease of development. Asp.NET applications can follow the more traditional web form paradigm, which utilizes event based programming.

## *Web Forms*

Using .NET there are two ways to develop robust interactive applications. The first and most traditional method is using the tried and true approach of web forms. Web forms are

comprised of a visual portion (an .aspx file), and code which is tied to actions and objects in the form, which are stored in a separate class file. Web forms serve to separate the HTML interface from the application's logic, and also allow programmers to write less code due to server-side .NET controls. Web forms also support Event- based programming, which is a commonly familiar ability.

### The MVC framework

**Models:** Models are the part of the application which implements the logic for the data. One of the common functionalities of a model object is to retrieve, store and update information with a database. In context of the Student Fee Committee, the model might retrieve information from a database table and then provide updated information back to the table.

**Views:** Views store the programming logic which dictates the application's user interface (UI). The UI is usually based on the model data. An example of this would be a proposal edit page which provides drop down lists, text boxes, and check boxes based on the state of the proposal.

**Controllers:** Controllers are essentially the brain behind the application. They control user interaction, interface with the model, and select which view to render. The view only displays information, the controller actually responds to the user's input. For instance, the controller takes strings as a query, and passes the values to the model. The model might in turn use this value to perform a query on the database.

The MVC pattern inherently separates different aspects of the application, while also providing ways for these elements to interact. If something is exceedingly difficult to program, it's very likely you're trying to do it in the wrong place (IE taxation calculations in the model).

The following diagram shows how information is passed between the models, the views, and the controller[1]. The controller holds the application's core logic.

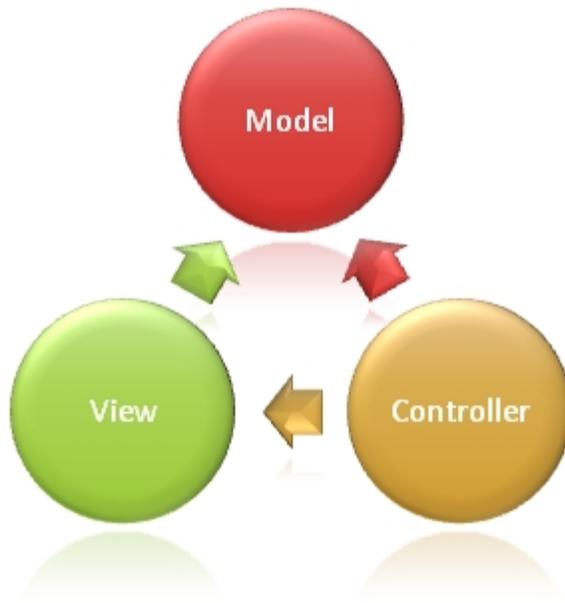


Figure 1: Model View Paradigm

Every language we investigated had a MVC based framework. VB and C# share the ASP.NET MVC framework, Ruby has a framework called Ruby on Rails, and Python has a framework called Django. Developers have recently been shifting over to this model due to the separation of concerns, and the inherent inclusion of the don't repeat yourself methodology.

## Human Factors Considerations

The current SFC proposal system requires a significant amount of human interaction. The human computer interface could be streamlined significantly for the process. Proposals wait for a human to act on them at 3 points in the current process. These points are listed below:

- 1) After the SFC admin Emails out the blank proposal form
- 2) After the proposer Emails the completed proposal back to the admin
- 3) Waiting to be printed out before proposal review meeting

By overhauling the SFC proposal system, this will be reduced to 2 periods waiting for human interaction. The first wait is after the SFC admin Emails out a notification to fill out a proposal. The second wait is waiting for the admin to print out the proposals before the meeting.

## Platform Constraints

The platform on which we could actually host a production application was one of the major hang-ups in our project. When things got closer to the due date, we decided that any functional application would suffice. The final selection was an ASP.NET C# application using traditional web form technology

## Database

### Defining Requirements

It is important when building a database and interface that the requirements come straight from the customer and you are solving the problem the customer has. Requirements

came from Joseph Anderson and Stephanie Allen. The requirements determined the tables which would be included in database.

The final requirements primarily consisted of a location to store the proposals themselves, the committee members and login information for them to use on the website. The database would need to have values inserted in through the web interface as well as report queries depending on what people were searching for. Additionally, we wanted to add consolidation features, so a report readout would be needed as well. Microsoft Access was chosen as the database management software to use here because of our past experience in it, the powerful report creation tools and its integration with Microsoft Visual Studio and ASP.NET and C#.

**Economic Analysis**

This project's economic success can be measured in time saved between the processes required in the original method compared to the new interface. The current method can be summarized by the following process chart:

<b>Current SFC Proposal Process(per Qtr)</b>	<b>estimated time(min)</b>
SFC Chair emails out proposals	10
SFC Chair compiles/sends proposals	20
Admin Coordinator resends to SFC board	15
SFC board receives/reviews	30
SFC reviews with proposers at meeting/recompile	35
Notes/decisions are compiled	10
Funds are allocated	45
Website is updated with current/past pending/approved	120
<b>Total</b>	<b>285</b>

Figure 2: Current SFC Proposal Process

A web interface would automate some of these recurring manual processes. The proposed process chart is as follows:

<b>Proposed SFC Proposal Process(per Qtr)</b>	<b>estimated time(min)</b>
Website is customized	5
SFC board receives/reviews	30
SFC reviews with proposers at meeting/Decisions are input	35
Funds are allocated	45
<b>Total</b>	<b>115</b>

**Figure 3: Proposed SFC Proposal Process**

Additional non-quantitative benefits may also be seen with the new web interface. Automated history can help students see all the proposals that have been accepted or rejected, even the most recent. PDF composition of all records will help track the data in paper form as well as improve consistency in formatting. Additionally, meeting minutes can be input directly by any member during the meeting. This as well as increased usability may encourage students to bring proposals forward. From a data management perspective, this means it will require less training to keep the SFC proposal system maintained.

## **Methods**

### **Web Interface Design**

The main consideration of the interface design was physical layout of form items. We decided that a good way to optimize this was to take notes from sites which handle form interactions seamlessly. The layout of the forms pretty much fell into place after a bit of observation.

## Database Design

The methodology behind our database design comes primarily from IME 312 (Database Design). We needed to create a simple, but powerful table structure that could be easily understood by future committee members for maintenance and future expansion. The database also needed to have no limitations as far as content being stored.

## Proposals

SFC Proposals are very formal and it is important that the data be stored safely and without redundancies. Due to the customer requests, proposals will carry an ID in the format of "4 digit school year"\_"counting integer". For example, 0809\_01, 0809\_02, as they currently are on the SFC webpage. This will become more difficult when inserting into the table, but for design can stay as type "Text" and also serve as Primary Key[4]. The Data Field names and Data Types are as follows:

	Field Name	Data Type
🔑	ID	Text
	Title	Text
	Problem	Memo
	Solution	Memo
	Request	Currency
	Author	Text
	Date	Date/Time
	Status	Text
	Attachments	Attachment

Figure 4: Proposals Table Field Data

Because each proposal has a set of items listed on it, an additional "ItemList" table[5] was linked to this one. It contains the ID as a foreign key as well as "ItemNumber" which is also a key. This will allow there to be multiple items associated with an individual ID. It is necessary to do it this way because there could be as few as 1 items and as many as 10, but we don't

know how many until the proposal is submitted. The "ItemList" Data Field and Data Types are listed below:

	Field Name	Data Type
🔑	ID	Text
🔑	ItemNumber	Number
	Description	Text
	TotalCost	Currency
	AmountRequested	Currency

Figure 5: Item List Table Field Data

The relationship between the tables is illustrated here:

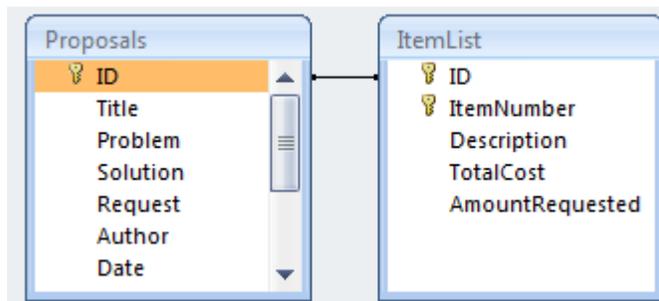


Figure 6: Proposals and Item List Relationship Structure

The Problems and Solution Fields were required to have the Data Type Memo, because text has a character limit, which might restrict length. Additionally, an Attachments Data Type was added for people who wish to attach pdf or other documents.

## Users

Currently, the SFC web page is not up to date on the committee members, officers and faculty members. In order to counter this issue, simple dynamic tables were made to hold information about new members[7]. When a new group of members are appointed, the users can be input into the system which will simultaneously create them login information based on their email(guaranteed unique, thus the ID) and update the tables being presented on the

home web page. This will keep the main page up to date as well as maintain the current login information. Login permissions are given based on standing(member, officer, advisor). The Table Relationship Structure is shown here:

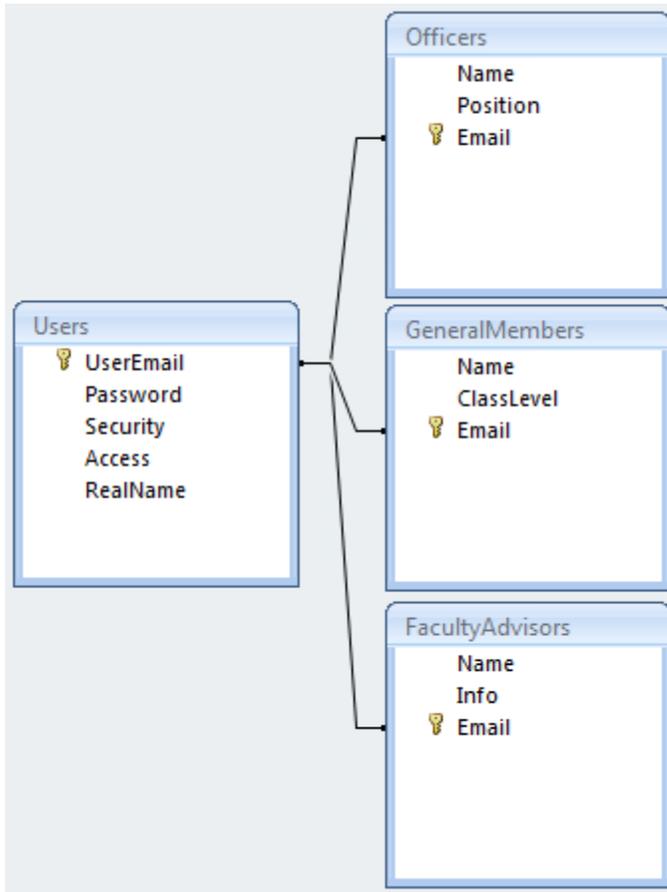


Figure 7: Members and Login Relationship Structure

## Report

SFC Proposals are important and need to be compiled in multiple places for redundancies. In order to aid in this, an Access Report was created to display the contents of each proposal, just as if it was submitted in document form. These reports can be printed, or exported into an individual pdf file[8]. This is a very valuable tool because compilation accounted for a very large portion of time required to maintain SFC Proposals. The report format is as follows:

Industrial and Manufacturing Engineering Department Fee Allocation Request Form

**SME SEMA Conference**

**To: IME Student Fee Committee**  
**From: Bobby Peterson, President**  
**Date: 10/9/2008**

**PROBLEM:**

IME students belonging to the SME club would like to attend the SME SEMA Conference in Las Vegas, Nevada, November 6th - 8th. This is the years manufacturing expo where vendors and companies all come to show their new products and companies advertise for employment.

**SOLUTION:**

This would give SME members the opportunity to explore potential employment opportunities and see the latest in manufacturing technologies.

**TOTAL COST OF PROJECT:**

Description of Item	TotalCost	AmountRequested
SME SEMA Conference Registration for 14 members	\$240.00	\$240.00
Lodging for 14 people Courtyard Marriot 3 rooms for 3 days	\$810.00	\$810.00
Rental vans & Gas	\$950.00	\$950.00

**Total:** \$2,000.00

Figure 8: Proposals Report Printout

The document sent out to students can be found below:

**Industrial & Manufacturing Engineering Department Fee Allocation Request Form**

**Title**

**To: IME Student Fee Committee**  
**From: Name**  
**Date: March 7, 2011**

**PROBLEM:**  
Problem

**SOLUTION:**  
Solution

**TOTAL COST OF PROJECT:**

Description of Item <small>(include manufacturer and model numbers if applicable)</small>	Total Cost	Amount Requested From Fee Committee
	\$0.00	\$0.00
<b>TOTALS</b>	<b>\$0.00</b>	<b>\$0.00</b>

Figure 9: Previous Proposals Submission Template



consistency, but also visibility. Visual Studio and Access 2011 are rumored to have the capability of posting reports directly to ASP.NET web forms. For now, authorized users can download the current build of the access database and use the report view there.

Another one of the key pages is the view past proposals page. The format of this page is as follows. We feel that the information is conveyed in a clear and straightforward manner.

**CAL POLY** [ [Log In](#) ]

**IME** Industrial and Manufacturing Engineering

Home | [Submit Proposal](#) | [Past Proposals](#) | [Meeting Minutes](#) | [Return to IME Website](#)

**PAST PROPOSALS**

Search by [Show All](#)

ID	Title	Author	Date	Status
0809_07	Travel IMAPS 2008	Krist	10/9/2008	Approved - Half
0809_06	Travel Support - Develop Sr. Project in Ghana	Gardner	10/9/2008	Tabled
0809_05	Travel for IME 101 Industry Tours	Karen Bangs	10/9/2008	Approved
0809_04	IME 429 Tape Measurers & Calipers	Liz Schlemer	10/9/2008	Approved
0809_03	IME 101 Industry Tours	Liz Schlemer	10/9/2008	Approved
0809_02	Tape Measurers For IME 443	Liz Schlemer	10/9/2008	Approved
0809_01	SME SEMA Conference	Bobby Peterson, President SME	10/9/2008	Approved
0809_14	Travel Support - Develop Sr. Project in Ghana (Resubmit)	Gardner	12/5/2008	Approved
0809_13	Textbooks for IME Sustainability council	Worth	12/5/2008	Approved
0809_12	IME 326 Microsoft Access Licenses	Lee & Yang	12/5/2008	Approved
0809_11	Student Assistant Technician for labs, Bldg. 41	Waldorf	12/5/2008	Approved
0809_10	Travel IIE/SME Tourfest 2009	Coogan & Otsuji	12/5/2008	Approved
0809_09	Travel Reimbursement for Balance of IMAPS Conference	Krist	12/5/2008	Approved
0809_08	Travel Reimbursement for SWE Conference	Harris	12/5/2008	Approved
1011_01	Reimbursement for Passing EIT Test	Ken Cairns	2/25/2011	Pending Review

To print proposals, open database in Access: [AccessDB](#)

Figure 11: Proposed View Consolidated Proposals Page

In total, the Access database has proven to be very useful.

## **Economic Analysis**

The values for estimated time in this analysis come from interviews and our estimations(Allen 2010). The current method has an estimated quarterly requirement of 4.75 hours and the proposed method has an estimated quarterly requirement of just under 2 hours. The difference in 2.75 hours per quarter gives us 8.25 hours saved per year, not including summer quarter. These savings will help the Administrative Coordinator use this time for more important tasks. In manual tasks like this, long intervals of time can go by without any updates(MESFAC). Our proposed method will help to keep even the most current proposals transparent.

## **Conclusion**

Although our final product is still not complete, we've learned a great deal about the workings of web 2.0 technologies. We've learned a significant amount about the embedding of programming languages into HTML, as well as more advanced concepts such as encapsulating coding logic and pseudo programming through interpreted terminal commands.

The relational database aspect was quite honestly one of the easiest concepts to grasp, based on our previous experience. We spent a significantly larger portion of time researching programming languages and frameworks, so that when we settled upon a solution, we were sure it would be appropriate.

## Works Cited

Ahlemann, Frederik. Towards a Conceptual Reference Model for Project Management Information Systems. Rep. 1st ed. Vol. 27. International Journal of Project Management, 2009. 14 Mar. 2008. Web. 09 Nov. 2010.

Allen Stephanie. "Student Fee Committee Website Scope." Personal interview. 29 Oct. 2010.

Callahan, John, and Brian Moretton. Reducing Software Product Development Time. Rep. 1st ed. Vol. 19. International Journal of Project Management, 2001. 27 Oct. 2000. Web. 9 Nov. 2010.

Gountanis, Chris. "Written Works." Chris Gountanis's Information Portal. Spring 2008. Web. 09 Nov. 2010. <<http://www.chrisgountanis.com/written-works.html?start=10>>.

Gray, Clifford F., and Erik W. Larson. Project Management: the Managerial Process. Boston: McGraw-Hill/Irwin, 2003. Print.

Hill, J., L. C. Thomas, and D. E. Allen. Experts' Estimates of Task Durations in Software Development Projects. Rep. 1st ed. Vol. 18. International Journal of Project Management, 2000. 11 Aug. 2000. Web. 9 Nov. 2010.

Huq, Faizul. Testing in the Software Development Life-cycle: Now or Later. Publication. 4th ed. Vol. 18. International Journal of Project Management, 2000. 5 Apr. 2000. Web. 09 Nov. 2010.

Khang, Do Ba, and Yin Mon Myint. Time, Cost and Quality Trade-off in Project Management: a Case Study. Publication. 4th ed. Vol. 17. International Journal of Project Management, 9 Feb. 1999. Web. 09 Nov. 2010.

Leahul, Dan. How To... Create a Website That Really Works. Rep. Revolution, Nov. 2009. Web. 9 Nov. 2010.

ME Department. "Mechanical Engineering Student Fee Committee Website." MESFAC. Jan. 2006. Web. 09 Nov. 2010. <<http://mesfac.calpoly.edu/>>.

Mehas, Nicholas. "Facility and Database Design for the Research Development Center." California Polytechnic State University Senior Project, March 2010. Web. 4 Nov. 2010.

Petter, Stacie. Managing User Expectations on Software Projects: Lessons from the Trenches. Rep. 7th ed. Vol. 26. International Journal of Project Management, 2008. 14 July 2008. Web. 9 Nov. 2010.

Sindhuja, P. N., and Surajith Ghosh Dastidar. Impact of the Factors Influencing Website Usability on User Satisfaction. Rep. IUP Journal of Management Research, Dec. 2009. Web. 9 Nov. 2010.

Sun, Jie, Wen Feng Lu, and Han Tong Loh. "Building a Database for Product Design Knowledge Retrieval - A Case Study in Robotic Design Database." Advanced Robotics and Machine Design 26.3 (2010): 224-29. June 2010. Web. 9 Nov. 2010.

Vicente, Kim J., and Jens Rasmussen. Ecological Interface Design: Theoretical Foundations. Rep. 4th ed. Vol. 22. IEEE, 1992. IEEE Transactions on Systems, Man, and Cybernetics. Web. 9 Nov. 2010.

Yang, Chao-Yang. "Website Designer as an Evaluator: A Formative Evaluation Method for Website Interface Development." Lecture Notes in Computer Science 5610 (2009): 372-81. Web. 9 Nov. 2010.

Yang, Tao. Data Management and System Design. Vol. E. San Diego: University Readers, Spring 2010. Print.

## Appendix A (Figures)

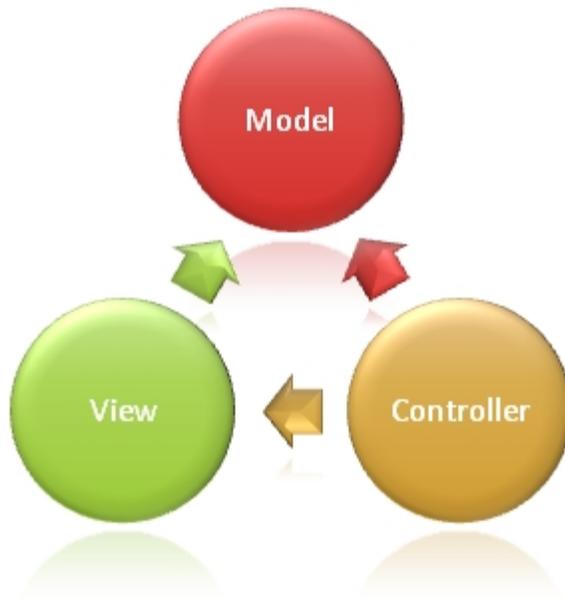


Figure 1: Model View Paradigm

Current SFC Proposal Process(per Qtr)	estimated time(min)
SFC Chair emails out proposals	10
SFC Chair compiles/sends proposals	20
Admin Coordinator resends to SFC board	15
SFC board receives/reviews	30
SFC reviews with proposers at meeting/recompile	35
Notes/decisions are compiled	10
Funds are allocated	45
Website is updated with current/past pending/approved	120
<b>Total</b>	<b>285</b>

Figure 12: Current SFC Proposal Process

Proposed SFC Proposal Process(per Qtr)	estimated time(min)
Website is customized	5
SFC board receives/reviews	30
SFC reviews with proposers at meeting/Decisions are input	35
Funds are allocated	45
<b>Total</b>	<b>115</b>

Figure 13: Proposed SFC Proposal Process

	Field Name	Data Type
🔑	ID	Text
	Title	Text
	Problem	Memo
	Solution	Memo
	Request	Currency
	Author	Text
	Date	Date/Time
	Status	Text
	Attachments	Attachment

Figure 14: Proposals Table Field Data

ItemList		
	Field Name	Data Type
🔑	ID	Text
🔑	ItemNumber	Number
	Description	Text
	TotalCost	Currency
	AmountRequested	Currency

Figure 15: Item List Table Field Data

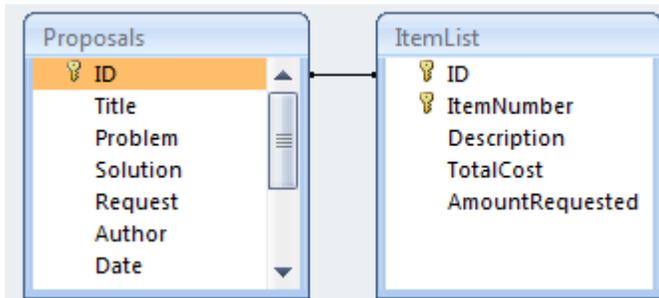


Figure 16: Proposals and Item List Relationship Structure

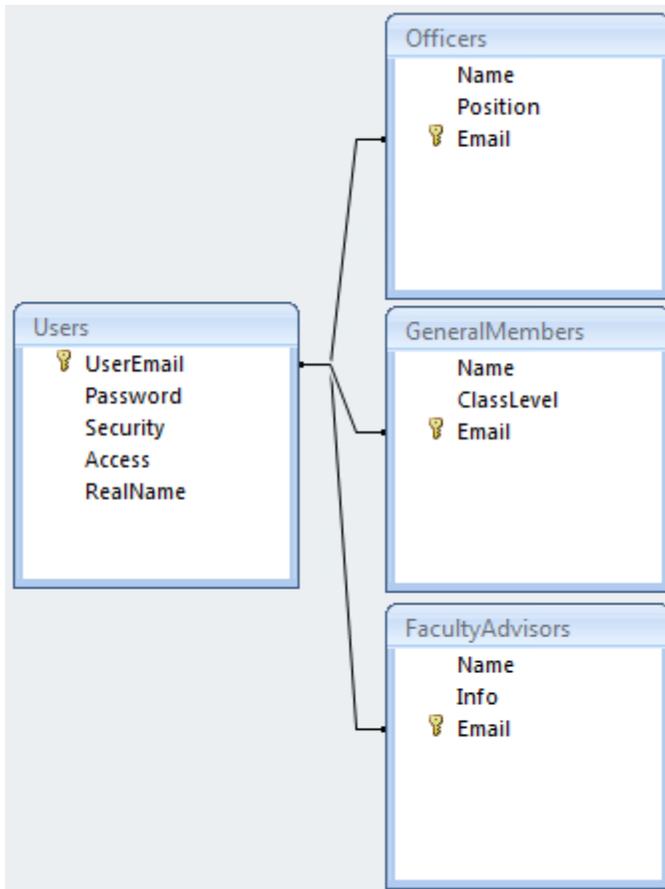


Figure 17: Members and Login Relationship Structure

Industrial and Manufacturing Engineering Department Fee Allocation Request Form

**SME SEMA Conference**

**To: IME Student Fee Committee**  
**From: Bobby Peterson, President**  
**Date: 10/9/2008**

**PROBLEM:**

IME students belonging to the SME club would like to attend the SME SEMA Conference in Las Vegas, Nevada, November 6th - 8th. This is the years manufacturing expo where vendors and companies all come to show their new products and companies advertise for employment.

**SOLUTION:**

This would give SME members the opportunity to explore potential employment opportunities and see the latest in manufacturing technologies.

**TOTAL COST OF PROJECT:**

Description of Item	TotalCost	AmountRequested
SME SEMA Conference Registration for 14 members	\$240.00	\$240.00
Lodging for 14 people Courtyard Marriot 3 rooms for 3 days	\$810.00	\$810.00
Rental vans & Gas	\$950.00	\$950.00

**Total:** \$2,000.00

Figure 18: Proposals Report Printout

Industrial & Manufacturing Engineering Department Fee Allocation Request Form

**Title**

To: IME Student Fee Committee

From: **Name**

Date: March 7, 2011

**PROBLEM:**

Problem

**SOLUTION:**

Solution

**TOTAL COST OF PROJECT:**

Description of Item <small>(include manufacturer and model numbers if applicable)</small>	Total Cost	Amount Requested From Fee Committee
	\$0.00	\$0.00
<b>TOTALS</b>	<b>\$0.00</b>	<b>\$0.00</b>

Figure 19: Previous Proposals Submission Template



**IME** Industrial and Manufacturing Engineering

- Home
- Submit Proposal
- Past Proposals
- Meeting Minutes
- Return to IME Website

**PAST PROPOSALS**

Search by

ID	Title	Author	Date	Status
0809_07	Travel IMAPS 2008	Krist	10/9/2008	Approved - Half
0809_06	Travel Support - Develop Sr. Project in Ghana	Gardner	10/9/2008	Tabled
0809_05	Travel for IME 101 Industry Tours	Karen Bangs	10/9/2008	Approved
0809_04	IME 429 Tape Measurers & Calipers	Liz Schlemer	10/9/2008	Approved
0809_03	IME 101 Industry Tours	Liz Schlemer	10/9/2008	Approved
0809_02	Tape Measurers For IME 443	Liz Schlemer	10/9/2008	Approved
0809_01	SME SEMA Conference	Bobby Peterson, President SME	10/9/2008	Approved
0809_14	Travel Support - Develop Sr. Project in Ghana (Resubmit)	Gardner	12/5/2008	Approved
0809_13	Textbooks for IME Sustainability council	Worth	12/5/2008	Approved
0809_12	IME 326 Microsoft Access Licenses	Lee & Yang	12/5/2008	Approved
0809_11	Student Assistant Technician for labs, Bldg. 41	Waldorf	12/5/2008	Approved
0809_10	Travel IIE/SME Tourfest 2009	Coogan & Otsuji	12/5/2008	Approved
0809_09	Travel Reimbursement for Balance of IMAPS Conference	Krist	12/5/2008	Approved
0809_08	Travel Reimbursement for SWE Conference	Harris	12/5/2008	Approved
1011_01	Reimbursement for Passing EIT Test	Ken Cairns	2/25/2011	Pending Review

To print proposals, open database in Access: [AccessDB](#)

Figure 21: Proposed View Consolidated Proposals Page