

CALNAIL – A DESIGN TOOL FOR SOIL NAIL PROJECTS USING FIELD CASE HISTORIES



FINAL REPORT
F/CA/IR-2006/05
JUNE 30, 2006
F-02-IR-16
EA 65-680531

CALNAIL – A DESIGN TOOL FOR SOIL NAIL PROJECTS USING FIELD CASE HISTORIES

**FINAL REPORT
JUNE 30, 2006
F/CA/IR-2006/05
F-02-IR-16
EA 65-680531**

**LOREN L. TURNER
JARED PARNELL**

CALIFORNIA DEPARTMENT OF TRANSPORTATION
DIVISION OF RESEARCH & INNOVATION
OFFICE OF MATERIALS & INFRASTRUCTURE
THE GEORESEARCH GROUP
5900 FOLSOM BLVD. MS-5
SACRAMENTO CA 95819
(916) 227-7174
LOREN.TURNER@DOT.CA.GOV

STATE OF CALIFORNIA DEPARTMENT OF TRANSPORTATION
TECHNICAL REPORT DOCUMENTATION PAGE
 TR0003 (REV. 10/98)

1. REPORT NUMBER F/CA/IR-2006/05	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
4. TITLE AND SUBTITLE CalNail – A Design Tool for Soil Nail Projects Using Field Case Histories	5. REPORT DATE June 2006	
	6. PERFORMING ORGANIZATION CODE	
7. AUTHOR(S) Loren L. Turner, Jared Parnell	8. PERFORMING ORGANIZATION REPORT NO. 65-339/65-680531, Rec. ID 0276	
9. PERFORMING ORGANIZATION NAME AND ADDRESS California Department of Transportation Division of Research & Innovation 5900 Folsom Blvd. MS-5 Sacramento, CA 95819	10. WORK UNIT NUMBER	
	11. CONTRACT OR GRANT NUMBER F-02-IR-16	
12. SPONSORING AGENCY AND ADDRESS California Department of Transportation Sacramento, CA 95819	13. TYPE OF REPORT AND PERIOD COVERED Final Report	
	14. SPONSORING AGENCY CODE	

15. SUPPLEMENTAL NOTES

16. ABSTRACT

Currently, geotechnical staff within Caltrans do not have ready access to detailed information on previously designed soil nail walls. Much of this information is kept by individual designers, or stored in paper format, with no organized method of retrieval. Having access to this institutional knowledge could help Caltrans carry out more effective designs. A wealth of soil nailing project data exists in literature as well as in state transportation agency records. Once compiled and organized this could serve as the basis of a high quality geotechnical information repository. This system has potential benefits to design engineers, consultants, contractors, planners, and researchers.

Deliverables resulting from the project include a design tool, called CalNail, and a database of soil nail wall case histories. CalNail is a web-based centralized repository of information for those involved in the design, construction, maintenance, and research of soil nailing infrastructure in California's highways. CalNail currently hosts over 30 documented case histories for soil nail walls in California.

17. KEY WORDS Soil nail walls, earth retaining structures, geotechnical data management systems	18. DISTRIBUTION STATEMENT No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161	
19. SECURITY CLASSIFICATION (of this report) Unclassified	20. NUMBER OF PAGES 110	21. PRICE

DISCLAIMER STATEMENT

THE CONTENTS OF THIS REPORT REFLECT THE VIEWS OF THE AUTHORS WHO ARE RESPONSIBLE FOR THE FACTS AND THE ACCURACY OF THE DATA PRESENTED HEREIN. THE CONTENTS DO NOT NECESSARILY REFLECT THE OFFICIAL VIEWS OR POLICIES OF THE STATE OF CALIFORNIA OR THE FEDERAL HIGHWAY ADMINISTRATION. THIS REPORT DOES NOT CONSTITUTE A STANDARD, SPECIFICATION, OR REGULATION. THIS REPORT DOES NOT CONSTITUTE AN ENDORSEMENT BY THE DEPARTMENT OF ANY PRODUCT DESCRIBED HEREIN.

TABLE OF CONTENTS

INTRODUCTION	1
BACKGROUND.....	1
EVALUATION OF THE ORIGINAL CALNAIL SYSTEM	2
DEVELOPING A NEW CALNAIL SYSTEM.....	9
SUMMARY	16
ACKNOWLEDGEMENTS	16

APPENDICES

APPENDIX A: SOFTWARE SYSTEM PROPOSAL

APPENDIX B: SOFTWARE PROJECT MANAGEMENT PLAN

APPENDIX C: SOFTWARE REQUIREMENTS SPECIFICATION

APPENDIX D: INSTALLATION OF CALNAIL

APPENDIX E: INSTALLATION OF MYSQL DATABASE FOR CALNAIL

APPENDIX F: INSTALLATION OF PHP FOR CALNAIL

APPENDIX G: CALNAIL USER'S GUIDE

INTRODUCTION

Currently, geotechnical staff within Caltrans do not have ready access to detailed information on previously designed soil nail walls. Much of this information is kept by individual designers, or stored in paper format, with no organized method of retrieval. Having access to this institutional knowledge could help Caltrans carry out more effective designs. A wealth of soil nailing project data exists in literature as well as in state transportation agency records. Once compiled and organized this could serve as the basis of a high quality geotechnical information repository. This system has potential benefits to design engineers, consultants, contractors, planners, and researchers.

Deliverables resulting from the project include a design tool, called *CalNail*, and a database of soil nail wall case histories. CalNail is a web-based centralized repository of information for those involved in the design, construction, maintenance, and research of soil nailing infrastructure in California's highways. CalNail currently hosts over 30 documented case histories for soil nail walls in California.

BACKGROUND

In 2002 a research project was initiated to develop a repository of data on the design and construction of Caltrans' soil nail walls. A web-based pilot information management system, called *CalNail*, was planned as a centralized source of information for those involved in the design, construction, maintenance, and research of soil nailing infrastructure on the state highway system. An initial version of CalNail was completed in mid 2003. However, with the departure of the Principal Investigator from Caltrans, the system was never fully test deployed, and further development work ceased. In July 2005 the project was reopened with a new Principal Investigator and project team. The products resulting from this renewed effort are presented in this final report.

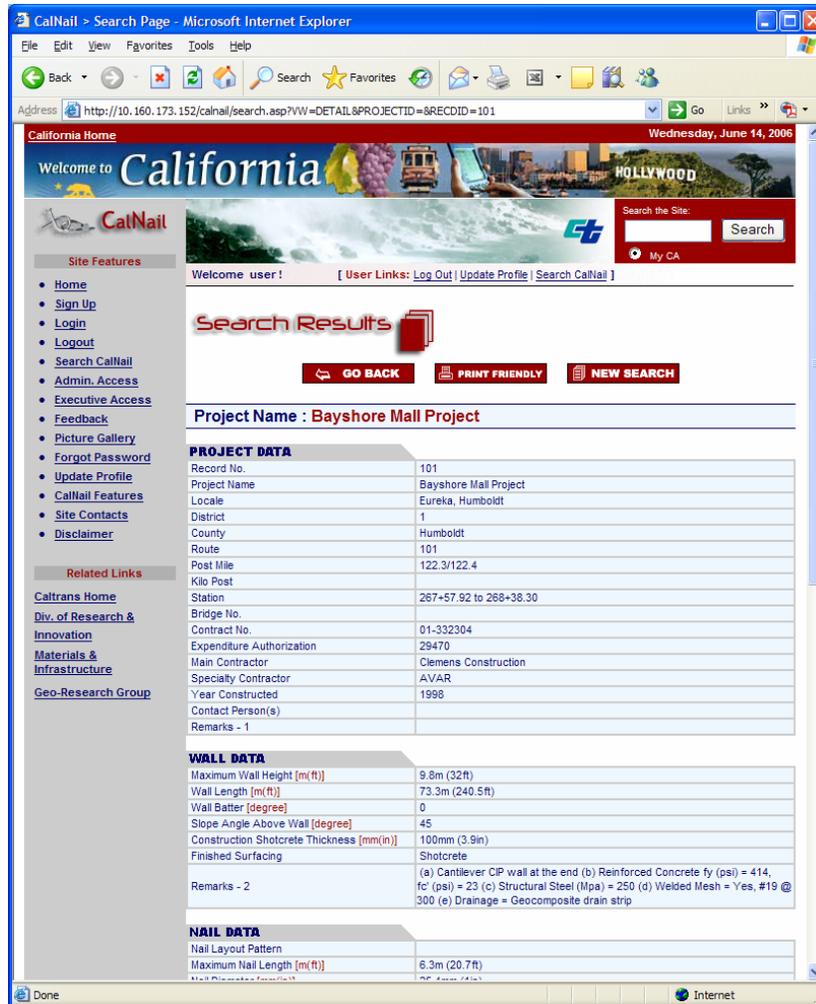
EVALUATION OF THE ORIGINAL CALNAIL SYSTEM

As a first order of work, the project team conducted a complete evaluation of the original CalNail system developed in 2003. The following aspects were considered in the evaluation:

Category	Evaluation
Database	<ul style="list-style-type: none"> ○ Does the data dictionary capture the essential data? ○ Is there sufficient metadata? ○ Are the data model and relational elements optimized?
Graphical User Interface	<ul style="list-style-type: none"> ○ Is the site navigation logical? ○ Are conventional controls, input, and navigational tools being used? ○ Can the user easily access information and data? ○ Does the site use consistent graphics and layout?
Functionality	<ul style="list-style-type: none"> ○ Are the searching tools effective? ○ Can the user access all available information? ○ Is user access to the system effectively managed?
Technology	<ul style="list-style-type: none"> ○ Is the system robust? ○ Do the technologies conform to Caltrans-approved IT standards? ○ What kind of resources are needed to support the system if deployed? ○ Are there any security issues?

The original CalNail system consisted of a database and a series of web pages that interacted with the database.





The database was intended to store information in an organized and logical format. Information associated with soil nailing projects was grouped into five broad categories:

- Project Data
- Wall Data
- Nail Data
- Soil Data
- Design and Testing Data

The Project Data entity contained information that primarily locates a project geographically and provides reference information. The Wall, Nail and Soil entities, in turn contained construction detail related information. The Design and Testing entity provided design and test related information pertaining to a project.

The original database model consisted of five tables, although, none of the tables had direct entity relationships. The tables were:

- Project_tbl
- Gallery_tbl
- User_tbl
- AutoEmail_tbl
- Feedback_tbl

Project_tbl contained all of the data associated with a particular soil nail. Each record in the table captured information about the wall, design, project, soils, tests, and nails. Many of the records contained redundant information since multiple walls often were associated with a single project. Gallery_tbl contained records for photos and a few attributes associated with each photo. However, the photo files were not directly associated with records in the Project_tbl. The User_tbl contained user account and subscription information. AutoEmail_tbl contained the content of automated email messages delivered by the system. Feedback_tbl contained records of comments submitted by users through the website. The overall structure of the database is shown below:

Project_tbl
ProjectID
Date
ProjectName
Locale
District
County
Route
PostMile
KiloPost
Station
BridgeNo
ContractNo
ExpAuth
MainContract
SpecialContract
YearConst
ContactPerson
ProjectRemarks
WallHeight
WallLength
WallBatter
SlopeAngle
ShotcreteThick
FinishSurface
WallRemarks
NailPattern
MaxLength
NailDiameter
NailInclination
VertSpace
HorizSpace
Centralizer
GroutDiameter
BarType
SteelGrade
Corrosion
NailRemarks
SoilType
FrictionAngle
Cohesion
SoilDensity
BoreholeLocation
Groundwater
BondStrength
SoilRemarks
DesignMethod
SurchargeLoad
TrafficLoad
SeismicLoad
AdditionalLoad
PunchShear
Software
TestLoad
TestNumber
DesignRemarks
Picture1
Picture2
Picture3

Gallery_tbl
GalleryID
Picture
Project
Contribution

AutoEmail_tbl
EmailID
EmailType
SenderAddress
Subject
Message

User_tbl
UserID
UserName
Password
FirstName
LastName
District
EmailAddress
PhoneNum
Privileges

Feedback_tbl
FeedbackID
Date
Name
Division
Comments
Remarks

Project_tbl consisted of 59 attributes associated with a particular soil nail wall as shown below.



Name	Type
ProjectID	Long Integer
Date	Date/Time
ProjectName	Memo
Locale	Memo
District	Memo
County	Memo
Route	Memo
PostMile	Memo
KiloPost	Memo
Station	Memo
BridgeNo	Memo
ContractNo	Memo
ExpAuth	Memo
MainContract	Memo
SpecialContract	Memo
YearConst	Memo
ContactPerson	Memo
ProjectRemarks	Memo
WallHeight	Memo
WallLength	Memo
WallBatter	Memo
SlopeAngle	Memo
ShotcreteThick	Memo
FinishSurface	Memo
WallRemarks	Memo
NailPattern	Memo
MaxLength	Memo
NailDiameter	Memo
NailInclination	Memo
VertSpace	Memo
HorizSpace	Memo
Centralizer	Memo
GroutDiameter	Memo
BarType	Memo
SteelGrade	Memo
Corrosion	Memo
NailRemarks	Memo
SoilType	Memo
FrictionAngle	Memo
Cohesion	Memo
SoilDensity	Memo
BoreholeLocation	Memo
Groundwater	Memo
BondStrength	Memo
SoilRemarks	Memo
DesignMethod	Memo
SurchargeLoad	Memo
TrafficLoad	Memo
SeismicLoad	Memo
AdditionalLoad	Memo
PunchShear	Memo
Software	Memo
TestLoad	Memo
TestNumber	Memo
DesignRemarks	Memo
Picture1	Memo
Picture2	Memo
Picture3	Memo
Video1	Memo

With the exception of the *ProjectID* and *Date* attributes, all other attributes were specified as a memo type, sometimes referred to as a string type. Attributes often associated with numeric

values (e.g. wall height) were stored as string values, making it challenging to conduct advanced searches (e.g. search for walls with heights between 20 to 25 meters).

The CalNail database was developed with Microsoft Access 2000 and was tested on a standalone Hewlett Packard Workstation loaded with Microsoft Windows 2000 Professional operating system configured with Internet Information Services (IIS) 5.0. The user would interact with the system using a series of web pages viewed with Microsoft Internet Explorer 5.0 or greater. The web pages were developed using Hypertext Markup Language (HTML), JavaScript, Stylesheets and Active Serves Pages (ASP). Access to the web pages was controlled with password authentication. In summary, the system featured:

- Error checking
- Password protection
- Picture uploading
- Printer friendly pages
- User registration
- Advanced searching

The evaluation of the original CalNail system identified a number of limitations and shortcomings that needed to be addressed in order for the system to be effective, extensible, and deployable at Caltrans. The evaluation criteria presented earlier in the document is presented again here along with further discussion.

Category	Evaluation
Database	<ul style="list-style-type: none"> ○ Does the data dictionary capture the essential data? <i>Yes. However, the data dictionary needs to be further expanded to facilitate capturing more detailed data if required. For example, one cannot document and associate boreholes with soil types and walls. Also, data is captured, for the most part, as text/string attributes. This makes data validation and advanced searching difficult or impossible to implement.</i> ○ Is there sufficient metadata? <i>No. As an example, location information is not captured in a structured format. One can input a station-offset without any information about the reference line, etc. One can input a lat/long without a datum.</i> ○ Are the data model and relational elements optimized? <i>No. The database model for the soil nail data consists of a single table, or "flat file". Separate tables exist only for user subscription data or feedback data. No entity relationships exist in the structure.</i>

<p>Graphical User Interface</p>	<ul style="list-style-type: none"> ○ Is the site navigation logical? <i>Yes and No. A user searching the system can quickly navigate to a single project. However, an administrator trying to enter new project data is led through a series of steps and interfaces that are not intuitive.</i> ○ Are conventional controls, input, and navigational tools being used? <i>Yes. Although, some elements of the navigation control are not entirely intuitive.</i> ○ Can the user easily access information and data? <i>Yes and No. The user can access the information available in the system. However, the Picture Gallery section will likely become unusable as more projects and photos are added to the system. Photos are not indexed to specific projects, which makes searching and browsing difficult.</i> ○ Does the site use consistent graphics and layout? <i>Yes. Graphics and layout are excellent.</i>
<p>Functionality</p>	<ul style="list-style-type: none"> ○ Are the searching tools effective? <i>No. The advanced search tool doesn't allow one to search by range of wall heights or range of soil conditions. This is likely the most common search criteria and is not adequately addressed.</i> ○ Can the user access all available information? <i>Yes. Although, photos associated with projects are difficult to identify without searching separately for them.</i> ○ Is user access to the system effectively managed? <i>Yes and No. Users can subscribe to the system very quickly. However, no user authentication is required in order to grant access.</i>
<p>Technology</p>	<ul style="list-style-type: none"> ○ Is the system robust? <i>Not sure. Acceptance testing was not conducted. In a limited test of the system, some database errors did come up. Microsoft Access will have limitations as to file size and record numbers. This could create problems in the future as more records are added.</i> ○ Do the technologies conform to Caltrans-approved IT standards? <i>No. The system uses Microsoft Access and ASP code. These technologies are not on the approved Caltrans IT list.</i> ○ What kind of resources are needed to support the system if deployed? <i>Not determined.</i> ○ Are there any security issues? <i>Not determined.</i>

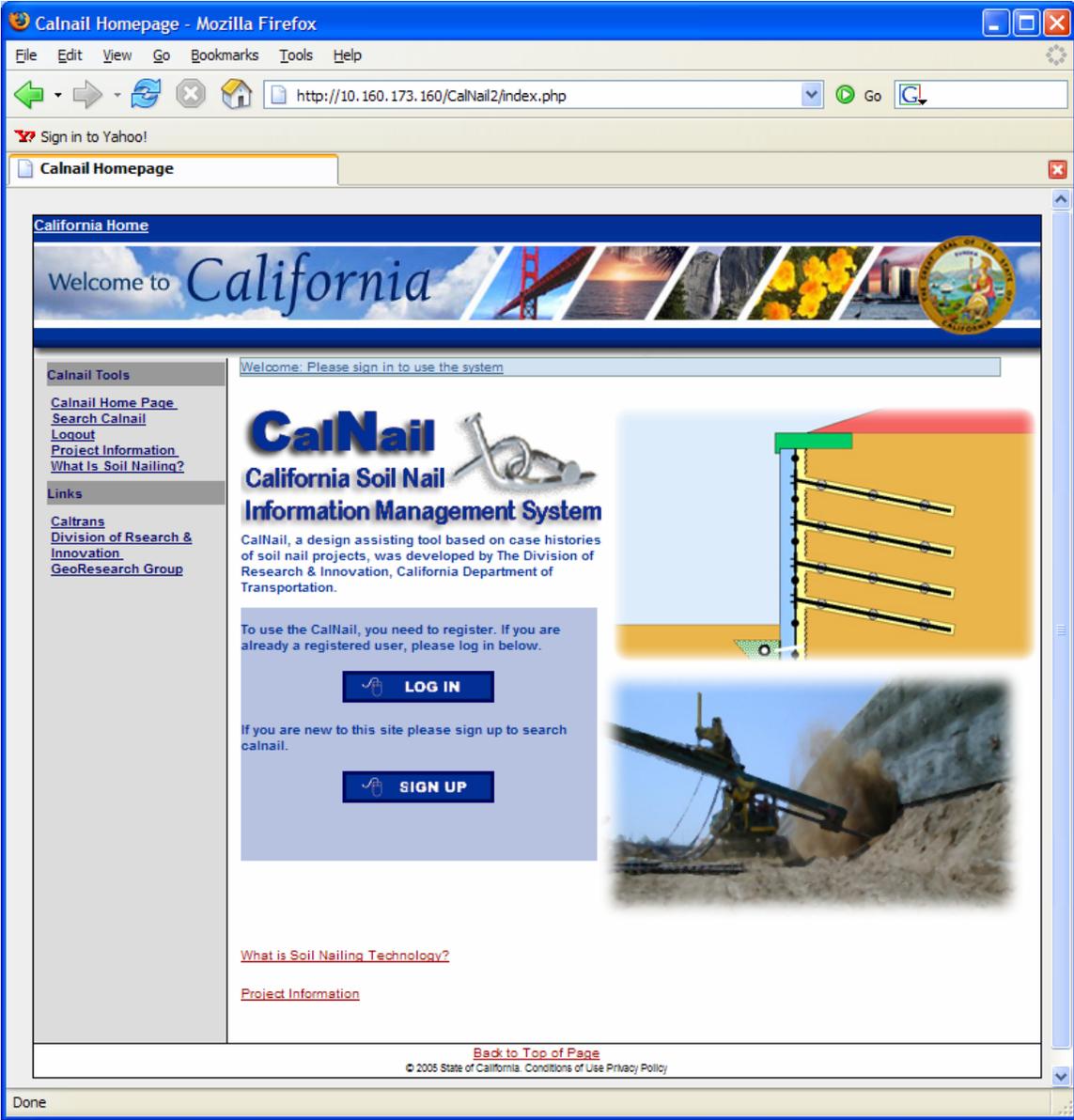
DEVELOPING A NEW CALNAIL SYSTEM

In July 2005 a meeting was held with the original project clients to present the findings of the evaluation of the original CalNail system. At this meeting the new project team presented a *Software System Proposal* (see Appendix A). The team proposed to address the shortcomings of the original CalNail system through a new development effort. In summary, this effort would include:

- Expand the data dictionary to capture comprehensive metadata and parameters.
- Restructure the database design utilizing a true relational model.
- Implement the database design in MySQL 4.0 (Caltrans approved standard).
- Utilize appropriate field types for different attributes.
- Develop new website navigation model and storyboard.
- Implement website tools, functions, and forms with PHP 5.0 (Caltrans approved standard).
- Enhance search functionality.
- Enhance file upload capability to store a index any file type.
- Implement more rigorous user subscription process with email validation.

This work was further documented in the *Software Project Management Plan* (see Appendix B) and the *Software Requirements Specification* (see Appendix C). Work on this project was conducted over a 10 month period by one undergraduate student intern. An estimated 800 hours (an equivalent of \$10,000) was expended for this project to account for time spent in development, coding, meetings, and documentation.

A beta version of the new CalNail system was unveiled to the project clients in May 2006. The new version retains all of the original wall and soil attributes and system features, in addition to some of the graphics, but implements a more extensible data model to accommodate complex types of data in a relational context as well as other significant improvements.

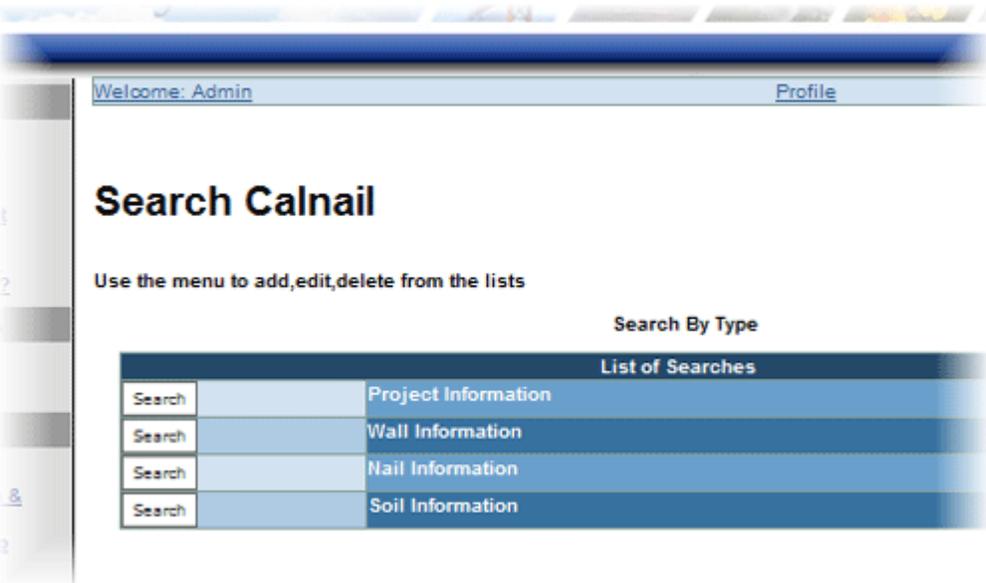


The revised version of Calnail features:

- Logical navigation tools -- constant links on sidebar and a "crumb trail" on top showing where you've visited.



- Advanced searching tools -- search by project, wall, nail, or soil attributes.



- Advanced searching tools – search by any parameter and/or range of parameters; use metric or English units.

Search by Wall

Search Wall Form			
Name	<input type="text"/>		
Length	<input type="text"/>	Meters	Equal to
Max Height	<input type="text"/>	Meters	Equal to
Batter Degree	<input type="text" value="(0.0-90.0)"/>		Greater Than
Upper Slope Angle	<input type="text" value="(0.0-90.0)"/>		Less Than
Facing Thickness	<input type="text"/>	cm	Greater or Equal to
Finish Description	<input type="text"/>		Less Than or Equal to
			Equal to
			Not Equal

Location Information	
Location Description	<input type="text"/>
LAT / LONG Coordinates	
LAT	Min: <input type="text"/> Max: <input type="text"/>
LONG	Min: <input type="text"/> Max: <input type="text"/>
Datum	Any

- Upload and store digital photos, video, and any other document type (e.g. SLOPEW files) with the project.

The screenshot displays a web browser window with a URL of `http://10.160.173.160/Calrail2/projects_edit.php?ProjectID=35`. The main content area is titled "Edit Project Form" and contains fields for Project Name, District, County, City, Route, and Project Completion. Below this is a "Files" tab with a table listing various files:

File Name	Description	Size	View	Edit	Delete
enr002.jpg	photo	0.10 KB	View	Edit	Delete
photoheader.jpg	photo	0.09 KB	View	Edit	Delete
multemore.jpg	photo	0.91 KB	View	Edit	Delete
walkwaylino.jpg	photo	0.08 KB	View	Edit	Delete
File Name	Description	Size	View	Edit	Delete
inspections.amsv.asp	Building Inspections	159 bytes	View	Edit	Delete
File Name	Description	Size	View	Edit	Delete
Design_Report.doc	DOC	0.02 KB	View	Edit	Delete
SHAL_analysis.BIML	SHAL analysis	11 bytes	View	Edit	Delete
Summary of Design Parameters.doc	Report	0.02 KB	View	Edit	Delete
WSLOPE_INPUT.DAT	WSLOPE	11 bytes	View	Edit	Delete

An inset window titled "projects_files_view.php [JPEG Image...]" shows a photograph of construction workers at a site. The browser's address bar for the inset window is `http://10.160.173.160/Calrail2/projects_files_view.php?table=Picture_Data&fileID=48&ProjectID=35`.

- Accommodate multiple walls, boreholes, nails, tests, and soils data within projects.

[Search](#) > [Search Results](#) > [View Project](#)

Project	
Project Name	Blue Hills
District	4 - Oakland
County	SCL - Santa Clara
City	Mountain View
Route	85
(Post Mile /Kilo Post)	- 4.1 / - 6.7 Miles
Bridge	37-0540M
EA Number	437554
Project Completion	1991-01-01
Comments	Associated Bridges: 37-540M 37-541M 37-542M 37-543M

Locations Contractors Contacts Files **Walls**

Name	Length	Height
Wall 1	304.8	
Wall 2	685.8	6.8
Wall 3	350.6	
Wall 4	929.7	6.8

- Utilize pull-down menus to select commonly input information (e.g. district, county, ASTM soil classifications, etc.).

Welcome: Admin [Profile](#)

[Project Management](#) > [Project: Turner Hill](#) > [Wall: Wall 1](#) > [Add Soil Data](#)

Fill in the blanks to add a new user
All RED fields are required

Add Soil Form	
Name	<input type="text"/>
Soil Type	<input type="text"/> <ul style="list-style-type: none"> (CH) - Fat clay (CV) - Clayey gravel (GC) - Clayey gravel with sand (GC-GM) - Silty, clayey gravel (GC-GM) - Silty, clayey gravel with sand (GM) - Silty gravel (GM) - Silty gravel with sand (GP) - Poorly-graded gravel (GP) - Poorly-graded gravel with sand (GP-GC) - Poorly-graded gravel with clay and sand (or silty clay and sand) (GP-GM) - Poorly-graded gravel with silt (GP-GM) - Poorly-graded gravel with silt and sand (GW) - Well-graded gravel (GW) - Well-graded gravel with sand (GW-GC) - Well-graded gravel with clay and sand (or silty clay and sand) (GW-GC) - Well-graded gravel with clay (or silty clay) (GW-GM) - Well-graded gravel with silt (GW-GM) - Well-graded gravel with silt and sand (MH) - Elastic silt (MH) - Elastic silt with gravel (MH) - Elastic silt with sand
Water Level	<input type="text"/>
Strength	<input type="text"/>
Density	<input type="text"/>
Cohesion	<input type="text"/>
Angle Phi	<input type="text"/>
Comments	<input type="text"/>

Submit Reset

- Automatic unit conversion -- user can choose to display data in metric or English units.

Fill in the blanks to add a new user
All RED fields are required

Add Soil Form	
Name	<input type="text"/>
Soil Type	(CH) - Fat clay
Water Level	20 Meters
Length	2 Meters
Density	kgs/m³
Cohesion	kPa
phi (Phi)	

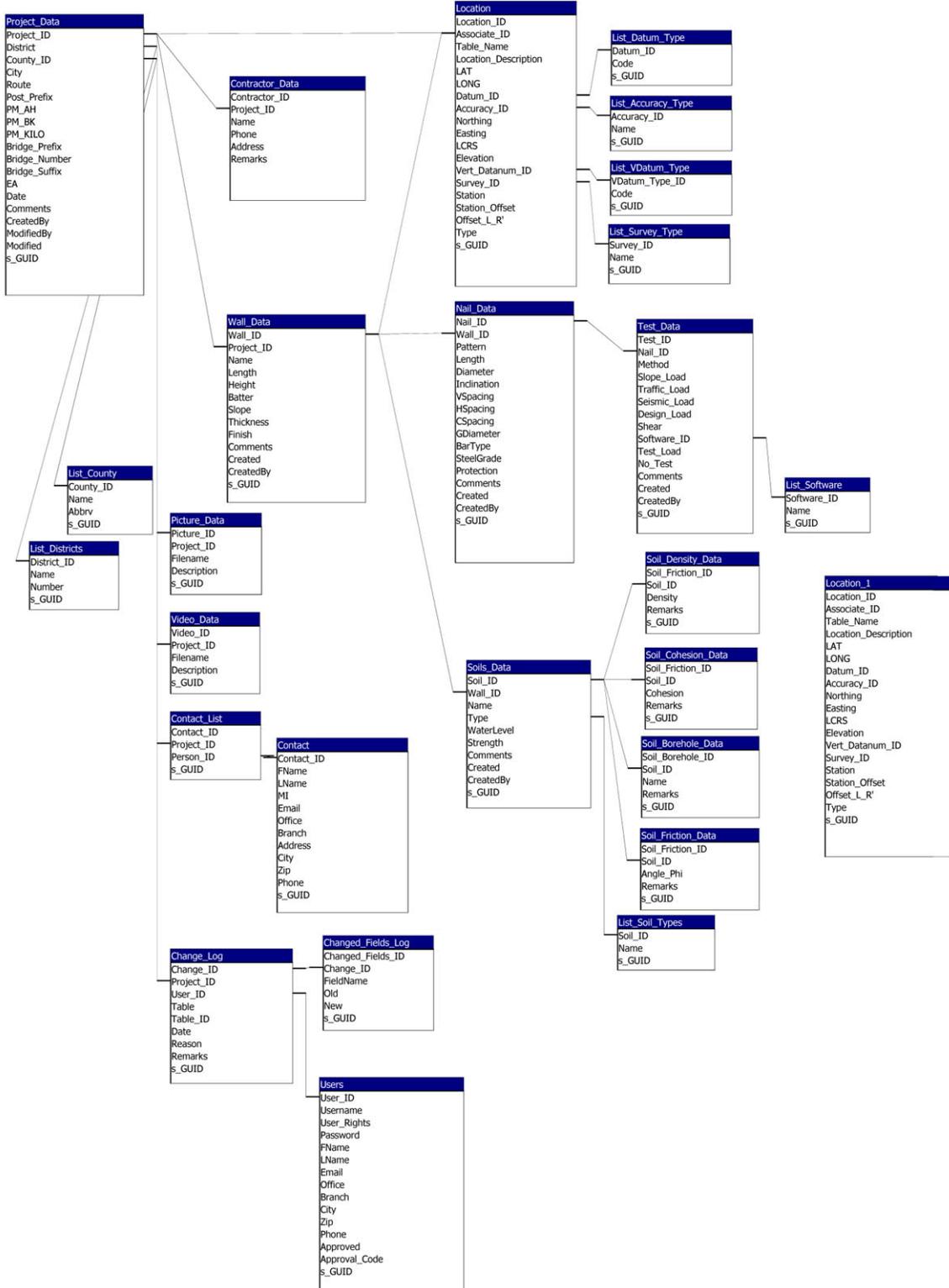
- Maintain user/subscriber profiles; Implements user validation -- new users register on the site and are sent an email with a validation link. A system administrator manages and validates new accounts.

Edit User

Fill in the blanks to add a new user
All RED fields are required

Edit User Form	
User Name	<input type="text" value="lturner"/>
Password	<input type="password" value="*****"/>
User Privileges	Administrator
First Name	loren
Last Name	turner
Email	<input type="text" value="loren.turner@dot.ca.gov"/>
Office	Materials and Infrastructure
Branch	GeoResearch Group
City	Sacramento
Zip	95819
Phone	916-227-7174
Approved	<input checked="" type="checkbox"/>

The revised database design reflects a true relational model. In the new model, separate tables are established for every element. As such, one can associate multiple soils, nails, and tests with walls. In turn, multiple walls can be associated with projects. Almost any element (e.g. wall endpoints, borehole locations, nail test locations, etc.) can have a spatial component identified by any number of locational methods (i.e. lat/long, Northing/Easting, station-offset). The database also incorporates a method for users to upload various digital media, includes photos, video clips, and other documents(e.g. Word, Excel, PDF, text, etc.), both proprietary and non-proprietary.



Complete documentation of the database model, data dictionary, and functionality can be found in Appendix G.

SUMMARY

A central repository of institutional knowledge of soil nail wall design and construction was needed for Caltrans. A research project was launched in 2002 to document this information through a web-based data repository. This initial effort was never fully tested and deployed due to the early loss of the original Principal Investigator and the development team. In 2005 and new development team was assembled with a new Principal Investigator. The system was completely redesigned from ground up. The beta product from this effort is the new CalNail system, as documented in this report. Further evaluation of the system by the end users is needed to assess deployment potential.

ACKNOWLEDGEMENTS

The authors would like to thank a number of people that have shared their insight and have made significant contributions to this project (in alphabetical order):

- John Ehsan (Geotechnical Design - South)
- Kathryn Griswell (Drilling Services)
- David Jang (Geotechnical Design - South)
- Hooshmand Nikoui (Geotechnical Design - West)

The authors would also like to acknowledge the efforts of the original development team:

- Ali Porbaha (Principal Investigator, Former Caltrans Transportation Engineer)
- Shyamal Gounder (Developer, Former Caltrans Student Assistant)

APPENDIX A:
CALNAIL
SOFTWARE SYSTEM PROPOSAL

SOFTWARE SYSTEM PROPOSAL

CALNAIL

TABLE OF CONTENTS

TABLE OF CONTENTS.....	2
1 INTRODUCTION.....	3
PROJECT HISTORY	3
PROJECT SPONSOR	4
DIVISION OF RESEARCH AND INNOVATION	4
1.1 PURPOSE	4
1.2 SCOPE	4
1.3 DEFINITIONS.....	4
2 PROJECT SPONSOR AND SPONSOR NEED.....	5
2.1 SPONSOR IDENTIFICATION.....	5
2.2 SPONSOR’S BUSINESS	5
Goals:.....	5
Values:.....	6
2.3 DESCRIPTION OF PROBLEM.....	6
3 SOLUTIONS.....	6
3.1 SOLUTION DESCRIPTION.....	6
4 TECHNICAL PROPOSAL	7
4.1 TECHNICAL DESCRIPTION OF THE PRODUCT DELIVERABLES	7
5 MANAGEMENT PROPOSAL	7
5.1 WORK SCHEDULE	7
5.2 RESOURCE REQUIREMENTS.....	8
5.3 COST	8
6 CONDITIONS AND COMMENTS.....	8
6.1 ASSUMPTIONS AND CONSTRAINTS.	8
6.2 GOALS.....	8
6.3 SUPPORT LIMITATIONS.....	8
6.4 OTHER DISCLAIMERS.	8
7 FEATURES OF ORIGINAL CALNAIL SYSTEM.....	9

1 INTRODUCTION

This is the Software System Proposal (SSP) for CalNail headed by the Division of Research and Innovation (DRI) for Caltrans as a research application.

The CalNail database is centralized source of information for those involved in the design, construction, maintenance and research of Soil Nailing infrastructure in California's highways. A wealth of Soil Nailing project data exists in literature and Department of Transportation (DOT) files. When compiled and organized this can serve as the basis of high quality geotechnical information repository. This system will benefit Design Engineers, Consultants, Contractors, Planners, and Research, and Development Professionals

PROJECT HISTORY

A research project was started in 2002 by Ali Porbaha (formerly in the division of Research and Innovation) and a group in the Geotechnical Services to develop a repository of data on design and construction of Caltrans' soil nail walls. A web-based pilot information management system, called "CalNail," was planned as a centralized source of information for those involved in the design, construction, maintenance, and research of soil nailing infrastructure on the state highway system. Once compiled and organized, CalNail would serve as the basis of a high quality information repository on soil nail walls, benefiting design engineers, consultants, contractors, planners, and research professionals.

An initial version of CalNail was completed in mid 2003. However, with Ali Porbaha's departure from Caltrans, the system was never fully test deployed, and further development work ceased.

The CalNail database was designed to store information in an organized and logical entity-table format. Information associated with Soil Nailing projects was grouped into five broad categories: Project Data, Wall Data, Nail Data, Soil Data, and Design and Testing Data. The Project Data entity contained information that primarily locates a project geographically and provided reference information. The Wall, Nail and Soil entities, in turn contain construction detail related information. The Design and Testing entity provided design and test related information pertaining to a project.

Original CalNail database architecture:

- Access 2000 Database
- ASP Pages
- Windows Server
- IIS 5.0

Original CalNail Features:

- Error Checking
- Password Protected
- Picture Uploading
- Printer Friendly
- Registration
- Searching / Advanced Querying
- See Section 7 for a complete List

PROJECT SPONSOR

John Ehsan
Geotechnical Design South
5900 Folsom Blvd Sacramento, CA 95819
(916) 227-7189
John_Ehsan@dot.ca.gov

DIVISION OF RESEARCH AND INNOVATION

Loren Turner
Jared Parnell
Division of Research and Innovation
5900 Folsom Blvd Sacramento, CA 95819
(916) 227-6975
Loren_Turner@dot.ca.gov

1.1 PURPOSE

The purpose of the Software System Proposal (SSP) is to provide a high-level understanding between the client and the development team on the product and what the needs of the product are. In addition the SSP will provide a means of contacting the development team.

1.2 SCOPE

The scope the SSP is that of an agreement between the DRI and our clients. The SSP serves as an affirmation that both parties, the development team and the sponsor, agree on the scope of the software project and outlines how tasks will be completed.

The SSP is a preliminary high-level understanding of the software problem. When new specifications, requirements, and constraints are discovered and defined they will be incorporated into the Software Requirements Specification (SRS). Should changes need to be made to the SSP once it has been base lined, management must approve and document the changes

1.3 DEFINITIONS

Acronym	Definition
.NET	The Microsoft .NET platform for developing web sites and programs
ASP	Active Server Pages
DOT	Department of Transportation
DRI	Division of Research and Innovation
FSR	Feasibility Status Report
MM	Maintenance Manual
MySQL	A open source SQL server

PHP	Pretext Hyperprocessor, a web based scripting language
SSP	Software System Proposal

2 PROJECT SPONSOR AND SPONSOR NEED

This section identifies the sponsor and the problem DRI will solve.

2.1 SPONSOR IDENTIFICATION

For more than 100 years, Caltrans and its predecessors have been responsible for planning, designing, building, operating and maintaining California's state highway system. Over time, that role has evolved to include rail and mass transit.

However, as California's transportation needs have broadened over the last century, so has Caltrans' focus. In the face of the state's burgeoning population, increased congestion and stubborn environmental pollution, Caltrans has moved to include new factors in its duties.

In addition to a changing mix of transportation modes - such as highways, rail, mass transit and aeronautics - Caltrans professionals today must consider such complex issues as land use, environmental standards, and the formation of partnerships between private industry and local, State and Federal agencies.

Caltrans today understands its purpose as promoting California's economic vitality and enhancing its quality of life by providing for the mobility of people, goods, services and information.

More specifically, Caltrans will continue to play its traditional role as owner and operator of the 15,000 mile State Highway System. Highways will continue as the backbone of the state's 'multimodal' transportation system. The Department is responsible for delivering the State's multibillion-dollar State Transportation Improvement Program.

2.2 SPONSOR'S BUSINESS

The California Department of Transportation strives to be the highest performing transportation agency in the country. In pursuit of our mission, we continue to build a talented and diverse team and to strengthen ties with our partners. To keep California moving, we commit ourselves to these goals and values.

GOALS:

- SAFETY
 - Provide the safest transportation system in the nation for users and workers
- RELIABILITY
 - Provide dependable travel times
- PERFORMANCE
 - Optimize transportation system throughput
- FLEXIBILITY
 - Provide mobility choices through strategic partnerships
- STEWARDSHIP

- Preserve and enhance California's resources and investments
- DELIVERY
 - Improve delivery of projects and services

VALUES:

- COMMITMENT
- COMMUNICATION
- CUSTOMER-FOCUS
- EMPOWERMENT
- INNOVATION
- INTEGRITY
- PARTNERSHIP
- STEWARDSHIP
- TEAMWORK

2.3 DESCRIPTION OF PROBLEM

CalNail suffers a variety of problems with its current web interface:

- Database structure unformatted
- Database Limited (e.g. Access)
- Difficult to update
- Fields not captured properly
- Flat file structure
- Inefficiencies in code
- Search ability Limited
- Site tough to navigate
- Code not compliant with Caltrans IT standards (e.g. ASP)

3 SOLUTIONS

3.1 SOLUTION DESCRIPTION

As a solution to the problems listed in the prior section DRI plans to do the following:

- Code a PHP application to use the updated database
- Create multiple search pages to take advantage of relational database
- Restructure the database using a relational model in MySQL
- Update field types with standard database practices
- Update site navigation to be more user friendly
- Use standard programming techniques to ensure code longevity

This solution allows even more potential to be added to the system by future programmers by creating a solid foundation on which new items could be stored.

4 TECHNICAL PROPOSAL

4.1 TECHNICAL DESCRIPTION OF THE PRODUCT DELIVERABLES

The product will be test deployed on a DRI server for a period of 18 months during the transition phase of development and deployment. The following is a list of materials to be delivered at the conclusion of the project.

- CD-Rom containing the completed proposed software, source code, installation manual, user manual, and maintenance manual.
- Hard copy of the installation manual.
- Hard copy of the user manual.
- Hard copy of the maintenance manual.
- Draft FSR document
- Final Report on System Performance

5 MANAGEMENT PROPOSAL

5.1 WORK SCHEDULE

This section describes the project deliverables and the delivery schedule. Each deliverable in the schedule has a start and end date, and an estimate of the hours necessary to complete each task

Deliverable	Start Date	End Date	Estimated Hours
Software Specification Proposal	7/5/05	7/15/05	40
Software Project Management Plan	7/18/05	7/25/05	40
Software Requirements Specification	7/25/05	8/8/05	80
Software Design Document	8/8/05	9/30/05	120
CODE	10/1/05	5/9/06	600
Software Testing Document	5/10/06	5/25/06	40
Software Testing Results	5/26/06	6/1/06	40
User Manual	6/2/06	6/10/06	24
MM	6/2/06	6/10/06	16
Final Report	6/10/06	END	40

Draft FSR	6/10/06	END	40
-----------	---------	-----	----

5.2 RESOURCE REQUIREMENTS

DRI will spend full work hours on this project until its completion unless some emergency event supercedes this project. All together roughly 540 hours will be spent researching, developing, and fine-tuning the web page to our clients specifications.

5.3 COST

Estimated costs are a result of manager wages plus developer's wages, these costs are estimates only and may not correspond to the actual incurred costs. All wages and hours are estimated. A detailed cost breakdown is available in the Project Management Plan.

Reimbursement rate: \$50/hr.

Estimated costs: \$500,000.00

6 CONDITIONS AND COMMENTS

6.1 ASSUMPTIONS AND CONSTRAINTS.

DRI assumes the client will cooperate with us in helping to gather requirements, point out issues and conflicts with their systems. We expect our clients to be willing to meet with the development team in order to find and resolve issues with the design of the system.

DRI assumes that once the design document is signed the product will not go through more design changes. Any additional changes will be incurred through a change request in which a new estimate of time will be given to show the new changes.

6.2 GOALS.

The goal for this product is a professionally designed web page with a stable easy to use backend that future developers will be able to maintain.

6.3 SUPPORT LIMITATIONS.

DRI support for this product is not unlimited; our test platforms are only available for the 18-month test deployment. We will not be able to support this product on a continual basis once it is deployed. It is up to the client to maintain its own web pages and server in the absence of DRI and its development team.

6.4 OTHER DISCLAIMERS.

This section shall remain blank until needed in the future.



Homepage

- Designed with guidelines from Governors Template.
- Navigation menu for ease of surfing.
- Related links to access various Caltrans websites.
- Contact information for database and website help.
- Timer based picture rotation for dynamic effect.
- Schematic illustration of Soil Nail Components.
- Project Information and acknowledgement.

User Sign-up

- Web interfaced registration.
- Instant access with registration.
- Personalized Username.
- Personalized Password.
- Updateable User Profile.
- Error handling for user sign-up.

User Login/Logout

- Security enabled with each session.
- No unauthorized use by non-registered members.
- Personalized greeting and links upon login.

Forgot Password Service

- Automatically generated email directly to you email mail account (web based).
- Error handling for forgot password service.

Picture Gallery

- Submit pictures.
- Upload pictures.
- View pictures.
- Print friendly option for clarity.

User Manual

- Online user manual.

User Feedback

- Submit new comments.
- View comments.
- Print friendly option for clarity.

Search Page

- Multiple Search criteria.
- Search engine based on:
 - Record No.
 - Project Name.
 - District.
 - County.
 - Route.
 - Contract No.
 - EA No.
- User-friendly drop down fields.
- Search criteria popup help.
- Custom navigation between searches.
- Colored California district map.
- Simplified viewing layout.
- Print friendly option for clarity.

Administration/Security Levels

- *User Privilege (Minimum access)*
 - General web site surfing privileges.
 - Project search privileges.
 - Profile update privileges.
 - Viewing comments privileges.
 - Submitting new comments privileges.
 - Uploading privileges.
 - Print privileges.
- *Administrator Privilege (Moderate access)*
 - General web site surfing privileges.
 - Project search privileges.
 - Profile update privileges.
 - Viewing comments privileges.
 - Submitting new comments privileges.
 - Uploading privileges.
 - Print privileges.
 - Viewing entire project information privileges (web based).
 - Adding new records to CalNail database privileges (web based).
 - Updating entire project information privileges (web based).
- *Executive Privilege (Unrestricted access)*
 - General web site surfing privileges.
 - Project search privileges.

- Profile update privileges.
- Viewing comments privileges.
- Submitting new comments privileges.
- Uploading privileges.
- Print privileges.
- Viewing records from various tables in the database privileges (web based).
- Adding new records to CalNail database privileges (web based).
- Updating records from various tables in the database privileges (web based).
- Deleting records from various tables in the database privileges (web based).

Database

- Access 2000 database.
- User table for user information.
- Project table for project information.
- AutoEmail table for automatically generated email information.
- Gallery table for picture gallery information.

APPENDIX B:
CALNAIL
SOFTWARE PROJECT MANAGEMENT PLAN

SOFTWARE PROJECT MANAGEMENT PLAN

CALNAIL

TABLE OF CONTENTS

TABLE OF CONTENTS	2
1 INTRODUCTION	4
PROJECT SPONSOR	4
DIVISION OF RESEARCH AND INNOVATION	4
1.1 PURPOSE	4
1.2 SCOPE	4
1.3 DEFINITIONS	5
1.4 REFERENCES	5
1.5 OVERVIEW OF CONTENTS OF DOCUMENT	6
2 PROJECT OVERVIEW	7
2.1 PROJECT SUMMARY	7
2.2 PROJECT DELIVERABLES	7
2.3 EVOLUTION OF THE SPMP.....	7
3 PROJECT ORGANIZATION	8
3.1 PROCESS MODEL	8
3.2 ORGANIZATION STRUCTURE AND INTERFACES.....	8
<i>Project Manager</i>	8
<i>Development Team</i>	8
<i>Document Organization</i>	9
3.3 ORGANIZATION AND STAFF.....	9
4 PROJECT MANAGEMENT AND CONTROL	10
4.1 PROJECT MANAGEMENT OBJECTIVES AND PRIORITIES.....	10
4.2 ASSUMPTIONS, DEPENDENCIES, AND CONSTRAINTS.....	10
<i>DRI assumes that:</i>	10
<i>Dependencies associated with the project:</i>	10
<i>Software and Platforms:</i>	10
<i>Time Constraints</i>	11
4.3 RISK MANAGEMENT	11
4.4 CHANGE MANAGEMENT	11
4.5 SCHEDULE CONTROL.....	12
4.6 ISSUE RESOLUTION	12
5 TECHNICAL PROCESS	13
5.1 METHODS, TOOLS, AND TECHNIQUES	13
5.2 SOFTWARE DOCUMENTATION	13

5.3	DOCUMENTS	13
6	ACTIVITIES, SCHEDULE, AND BUDGET	14
6.1	ACTIVITIES AND TASKS	14
6.2	RESOURCE REQUIREMENTS	14

1 INTRODUCTION

This is the Software Project Management Plan (SPMP) for the development of CalNail, a research demonstration project led by the Division of Research and Innovation (DRI).

CalNail is a web-based database system that serves as a centralized source of information for those involved in the design, construction, maintenance, and research of soil nailing infrastructure on California's highways. A wealth of soil nailing project data exists in literature and Department of Transportation (DOT) files. When compiled and organized this can serve as the basis of high quality geotechnical information repository. This system will benefit design engineers, consultants, contractors, planners, and research, and Development Professionals

PROJECT SPONSOR

John Ehsan Chief, Geotechnical Design South-1 5900 Folsom Blvd Sacramento, CA 95819 (916) 227-7189 JOHN_EHSAN@DOT.CA.GOV

DIVISION OF RESEARCH AND INNOVATION

Loren Turner GeoResearch Group Division of Research and Innovation 5900 Folsom Blvd Sacramento, CA 95819 (916) 227-7174 Loren_Turner@dot.ca.gov
Jared Parnell GeoResearch Group Division of Research and Innovation 5900 Folsom Blvd Sacramento, CA 95819 (916) 227-6975

1.1 PURPOSE

The purpose of the SPMP is to outline the tasks of developing, delivering, and implementing the CalNail system for Caltrans, and to document how the system will be managed. It defines the tasks and a schedule for completion of the CalNail system.

1.2 SCOPE

The scope of this document is the organization of the project. The project organization includes the management component of the project and the team member role and responsibility component. The document also outlines the tools and techniques that will be used to complete the project and the specific activities and tasks required to complete the project. Overall, this document centers on the management of these integral project components.

1.3 DEFINITIONS

Acronym	Definition
.NET	The Microsoft .NET platform for developing web sites and programs
ASP	Active Server Pages
DOT	Department of Transportation
DRI	Division of Research and Innovation
FSR	Feasibility Study Report
MM	Maintenance Manual
MySQL	An open source SQL server
PHP	Hypertext Preprocessor; General-purpose scripting language that can be used for web application development and can be embedded into HTML.
QA	Quality Assurance
SDD	Software Design Document: Document which describes in detail how project was implemented
SMM	Software Maintenance Manual: a technical maintenance manual that describes how to update and maintain a system after initial installation.
SPMP	Software Project Management Plan: a document that outlines the management of a software project.
SRS	Software Requirements Specification: a document which describes all the requirements related to the project.
STD	Software Testing Document: a document outlines the test procedures and strategies for the software
UM	User Manual

1.4 REFERENCES

Buckley, R., *Guide to Preparing the Software Project Management Plan*. Department of Computer Science – College of Engineering and Computer Science, California State University, Sacramento

1.5 OVERVIEW OF CONTENTS OF DOCUMENT

Section 2 - This section is an overview of the project. It provides a summary of the projects and lists the project deliverables. It also details how the Software Project Management Plan (SPMP) will evolve during the project lifecycle.

Section 3 - This section outlines the organization of the project. It includes a process model, the organizational structure of the team, the roles and responsibility of each team member.

Section 4 - This section outlines the manner the project will be managed. Management entails keeping the project up to date and current, measuring the progress of the project, maintaining the project schedule, developing the software, and verifying and validating the software. Resolving each of these issues will be explained in this section.

Section 5 - This section outlines all the documentation that will be produced and the dates of their completion throughout the project lifecycle. It also describes the tools, methods, and techniques required to complete the project. This section also contains the software documentation plan.

Section 6 - This section describes the activities and tasks that are essential for project completion. It also includes the schedule of deliverables and a proposed budget.

2 PROJECT OVERVIEW

This section contains an overview and summary of the project. Also included is a list of deliverables. The guidelines as to how the SPMP will be used, how it will be updated, and how it will evolve over the course of the project is also explained.

2.1 PROJECT SUMMARY

CalNail will incorporate the full model layout for software engineering as outlined in Computer Science. It will include the following development phases that are explained in Section 3.1:

- Sponsor and Project Identification
- Proposal
- Management Plan
- Requirements Specification
- Implementation
- Testing
- User and Maintenance Manuals
- Installation and Delivery
- Final Report
- Feasibility Status Report

2.2 PROJECT DELIVERABLES

- Software System Proposal (SSP)
- Software Project Management Plan (SPMP)
- Software Design Description (SDD)
- Code
- Software Testing Description (STD)
- Software Testing Report (STR)
- User's Manual (UM)
- Maintenance Manual (MM)
- Final Product
- Final Report
- Draft Feasibility Study Report (FSR)

2.3 EVOLUTION OF THE SPMP

The SPMP will serve as a baseline for our project. However, changes will be made to this plan as deemed necessary. Changes made to the SPMP are at the discretion of the development team and the sponsor. Changes will be announced at project meetings or through email and will be distributed to all team members as well as the project sponsor.

3 PROJECT ORGANIZATION

This section details each major function and activity associated with the project as well as the responsibilities of each member.

3.1 PROCESS MODEL

The software engineering process model that will be used throughout project is the iterative waterfall model.

As described in the process model, after identifying a sponsor and a project, the team will generate a Software System Proposal (SSP) to be approved by the sponsor. The Software Project Management Plan (SPMP) provides for the monitoring and control of the entire project. The Systems Requirements Specification (SRS) will convey the specific requirements of the project and enable both the sponsor and the development team to understand the solution to the project. Coding and testing the software product follows the SRS. A User Manual (UM) and a Maintenance Manual (MM) will be written. The UM will provide instruction on how to use the software while the MM will provide instruction on how to update and maintain the software.

3.2 ORGANIZATION STRUCTURE AND INTERFACES

This subsection discusses the team's organization structure.

Name	Role
John Ehsan, Geotechnical Design South 1	Project Sponsor
Loren Turner	Project Manager
Jared Parnell	Development Team

PROJECT MANAGER

The Project Manager will lead the team in accomplishing assigned tasks. The PM will be the driving force that keeps the project progressing and on schedule. The PM duties include:

- Assigning tasks
- Monitoring progress and assigning corrective actions
- The Project Manager is the final authority in all team decisions
- Quality Assurance for all deliverables
- Producing the Final Report
- Producing the draft FSR

DEVELOPMENT TEAM

The Development team is responsible largely in part for the coding and documentation of the project. They are responsible for producing, checking, and finalizing all documents and deliverables to the sponsor. Some of the responsibilities will include:

- Gathering Requirements
- Producing product mock ups for reviews
- Coding
- Documentation of code

- Produce Reports for the Project Manager

DOCUMENT ORGANIZATION

To ensure the quality assurance of our documentation, each document will be reviewed for problems throughout its development. Once the document is completed the Project Manager will review it and then return it with notes and comments for changes and revisions. The Development Team will then revise the document appropriately and return the documents to the Project Manager in a timely manner.

3.3 ORGANIZATION AND STAFF

This subsection describes the project's tasks and the team member(s) who will be the leads for each task.

Deliverables	Primary Lead	Secondary Lead
Software System Proposal (SSP)	Development Team	Project Manager
Software Project Management Plan (SPMP)	Development Team	Project Manager
Software Requirements Specification (SRS)	Development Team	Project Manager
Software Design Description (SDD)	Development Team	Project Manager
Code	Development Team	Project Manager
Software Testing Description (STD)	Development Team	Project Manager
Software Testing Report (STR)	Development Team	Project Manager
User's Manual (UM)	Development Team	Project Manager
Maintenance Manual (MM)	Development Team	Project Manager
Final Report	Project Manager	
Draft Feasibility Study Report (FSR)	Project Manager	

This section outlines how the project will be managed and controlled. It gives details on how the SPMP will be kept current, how the project progress is measured, how schedules will be tracked, what software development methodologies will be used, how verification and validation will be conducted. This section will also detail the plan for delivery, installation, and acceptance of the software product.

4.1 PROJECT MANAGEMENT OBJECTIVES AND PRIORITIES

The goal of the team from the Division of Research and Innovation (DRI) is to improve, develop, design, and implement CalNail, a web-based, soil nail case history, archiving system for users within Geotechnical Services.

Our first priority is getting the initial product requirements clearly defined so that development is not impeded by significant changes to the overall design in the middle of the development process. Requests to add or modify features once a baseline of requirements is established will go a change management process as described in Section 4.4.

4.2 ASSUMPTIONS, DEPENDENCIES, AND CONSTRAINTS

Several assumptions, dependencies and constraints involved in bringing this project to completion are outlined in this section.

DRI ASSUMES THAT:

- The sponsor will meet with the development team when necessary to aid development tasks such as requirements gathering and approval of layouts
- Sponsor will be willing to take an active part reviewing key material on the design of the product
- Sponsor will communicate concerns, questions, ideas, and assumptions to the development team so that the project lifecycle will flow smoothly

DEPENDENCIES ASSOCIATED WITH THE PROJECT:

- The sponsor and development team must agree on the precise functionality, interface and platform of the software before development begins.
- The development tools and procedures will be agreed upon before beginning the coding phase.

SOFTWARE AND PLATFORMS:

- MySQL will be used as the database management system
- PHP will be used for the web interface
- The platform will reside on a DRI test server for 18 months

TIME CONSTRAINTS

Some time constraints are outlined in the section 4.5 below. But the two most important dates are the completion of SRS and the project delivery and installation. During school session the development team will not be able to provide as much effort as during the summer break. School obligations will override normal development time if problems or emergencies occur.

4.3 RISK MANAGEMENT

As with all software development projects, there are risks associated with developing this software. The main risks that can derail the project are loss of a team member and loss of project.

If a loss of a team member occurs then replacement staff will have to fill in until project completion.

To help deter the loss of data a selected save period will be initiated to keep code backups both on the server and development workstations

To development resources become for some reason unavailable DRI will try to provide alternatives to

4.4 CHANGE MANAGEMENT

Change management is a process that provides a mechanism to identify and handle change in order to maintain the balance between requirements on one hand and the cost/schedule on the other. This process allows for change during the project's life cycle, but always puts the change in the context of the latest documented agreement (project plan) between the team.

The change management process consists of a series of steps that allows change to be identified, evaluated, priced, and tracked through closure. The goal is implement and use a process that fits the project.

- Step 1 – The proposed change is described which clearly identifies whether the change is system, organizational, or procedural in nature. Any reference material that will assist the reviewers will be identified and attached. A discussion of why the change is being proposed, including a cost-benefit analysis, if needed. If the change is not implemented, how will it adversely affect the customer and the state organization? Attach any supporting documentation that helps to clarify the proposed change. When complete, the change management document is submitted to the project manager.
- Step 2 - All change requests will be reviewed on a regular basis by the project manager, who will review the initial request and determine whether to proceed, reject, or defer the request. The project manager will make an initial assessment of the cost, schedule, and resources needed to implement the proposed change.
- Step 3 -- With project manager approval, the appropriate processes will be followed to update baseline documents.

When making a change to the software design or fixing problems with the software a report will be generated of the problem and include an issue resolution where the proposed solution will be reviewed by the project lead and approved.

4.5 SCHEDULE CONTROL

By comparing the number of estimated hours with the bi weekly timesheet the Project Manager will be able to track and monitor the progress and effectiveness of the project schedule as described in Section 6.1. The change management process, described in Section 4.4, will be used to make adjustments to the schedule.

4.6 ISSUE RESOLUTION

Project meetings will be used to solve conflicts in the event that two or more members have a conflict in ideas and/or issues. Meetings will be open for discussion by all members of the team, not just the conflicting parties. All members will give possible solutions to the problem, and the team will decide on the best solution to the problem.

For a problem concerning any part of the technical design phase, the issue will be brought up during the technical review. Here the problem will be presented to the whole team, and all members can give input on possible solutions. We will utilize the technical review to solve any problem that will occur during the technical process.

This section outlines all the documentation that will be produced and the dates of their completion throughout the project lifecycle. It also describes the tools, methods, and techniques required to complete the project. This section also contains the software documentation plan.

5.1 METHODS, TOOLS, AND TECHNIQUES

The development methodologies include the requirements development practices, design methodologies and notations. The language that will be used is PHP (interface) for the front end and MySQL(database) for the back end. Quality assurance practices will be included in our project by methods of technical peer reviews, unit testing, system testing, and regression testing.

5.2 SOFTWARE DOCUMENTATION

The documents that will be produced through out the life cycle of our project will go through the following process:

- Documentation will be written by the development team
- Review with the project manager
- Finalization of the document and a brief review
- Presentation for approval to sponsor if necessary

5.3 DOCUMENTS

The following documents will be produced over the development life cycle:

- Software Specification Proposal
- Software Project Management Plan
- Software Requirements Specification
- Software Design Document
- Software Testing Document
- Software Testing Results
- User Manual
- Maintenance Manual
- Final Report
- Draft Feasibility Study Report

6 ACTIVITIES, SCHEDULE, AND BUDGET

This section contains the Work Breakdown Structure for the project. Also included is a list of resource requirements, proposed budget, and a Gantt chart indicating a baseline schedule.

6.1 ACTIVITIES AND TASKS

Deliverable	Start Date	End Date	Estimated Hours
Software Specification Proposal	7/5/05	7/15/05	40
Software Project Management Plan	7/18/05	7/25/05	40
Software Requirements Specification	7/25/05	8/8/05	80
Software Design Document	8/8/05	9/30/05	120
CODE	10/1/05	5/9/06	600
Software Testing Document	5/10/06	5/25/06	40
Software Testing Results	5/26/06	6/1/06	40
User Manual	6/2/06	6/10/06	24
MM	6/2/06	6/10/06	16
Final Report	6/10/06	END	40
Draft FSR	6/10/06	END	40
Total Estimated Hours			1080 hours

6.2 RESOURCE REQUIREMENTS

Resources that might be needed include reference materials involving PHP and MySQL.

APPENDIX C:
CALNAIL
SOFTWARE REQUIREMENTS SPECIFICATION

SOFTWARE REQUIREMENTS SPECIFICATION

CALNAIL

Table of Contents

TABLE OF CONTENTS	2
1 INTRODUCTION	4
PROJECT SPONSOR.....	4
DIVISION OF RESEARCH AND INNOVATION	4
1.1 PURPOSE	4
1.2 SCOPE	5
1.3 DEFINITIONS.....	5
1.4 REFERENCES	6
1.5 OVERVIEW OF CONTENTS OF DOCUMENT	6
2 GENERAL DESCRIPTION	7
2.1 PRODUCT PERSPECTIVE.....	7
2.2 USE CASE DIAGRAM.....	7
2.3 USER CHARACTERISTICS	9
2.3.1 Administrators.....	9
2.3.2 Managers.....	9
2.3.3 Viewers	9
2.4 GENERAL CONSTRAINTS	9
2.5 ASSUMPTIONS AND DEPENDENCIES	10
3 SPECIFIC REQUIREMENTS	11
3.1 USE CASE SPECIFICATIONS.....	11
3.1.1 Login.....	11
3.1.2 Logout.....	11
3.1.3 Register.....	12
3.1.4 Search.....	12
3.1.5 Advanced Search	13
3.1.6 Manage Projects.....	13
3.1.7 Add Project.....	14
3.1.8 Edit Project.....	14
3.1.9 Delete Project.....	15
3.1.10 Manage Contractors.....	15
3.1.11 Add Contractor.....	16
3.1.12 Edit Contractor.....	16
3.1.13 Delete Contractor.....	17
3.1.14 Manage Contacts.....	17
3.1.15 Add Contact.....	18
3.1.16 Edit Contact.....	18
3.1.17 Delete Contact.....	19

3.1.18 Manage Pictures.....	19
3.1.19 Add Picture.....	20
3.1.20 Edit Picture.....	20
3.1.21 Delete Picture.....	21
3.1.22 Manage Files.....	21
3.1.23 Add File.....	22
3.1.24 Edit File.....	22
3.1.25 Delete File.....	23
3.1.26 Manage Walls.....	23
3.1.27 Add Wall.....	24
3.1.28 Edit Wall.....	24
3.1.29 Delete Wall.....	25
3.1.30 Manage Nails.....	25
3.1.31 Add Nail.....	26
3.1.32 Edit Nail.....	26
3.1.33 Delete Nail.....	27
3.1.34 Manage Tests.....	27
3.1.35 Add Test.....	28
3.1.36 Edit Test.....	28
3.1.37 Delete Test.....	29
3.1.38 Manage Soils.....	29
3.1.39 Add Soil.....	30
3.1.40 Edit Soil.....	30
3.1.41 Delete Soil.....	31
3.1.42 Manage Users.....	31
3.1.43 Add User.....	32
3.1.44 Edit User.....	32
3.1.45 Delete User.....	33

1 Introduction

This section details a summary of Division of Research an Innovation Software Requirements Specification, the SRS. This section will define the scope and the purpose of the SRS. It also defines key words, abbreviations, and acronyms that will be used throughout the document. A description of the contents of the document is also included in this section.

CalNail is a web-based database system that serves as a centralized source of information for those involved in the design, construction, maintenance, and research of soil nailing infrastructure on California's highways. A wealth of soil nailing project data exists in literature and Department of Transportation (DOT) files. When compiled and organized this can serve as the basis of high quality geotechnical information repository. This system will benefit design engineers, consultants, contractors, planners, and research, and Development Professionals

Project Sponsor

John Ehsan Chief, Geotechnical Design South-1 5900 Folsom Blvd Sacramento, CA 95819 (916) 227-7189 JOHN_EHSAN@DOT.CA.GOV

Division of Research and Innovation

Loren Turner GeoResearch Group Division of Research and Innovation 5900 Folsom Blvd Sacramento, CA 95819 (916) 227-7174 Loren_Turner@dot.ca.gov
Jared Parnell GeoResearch Group Division of Research and Innovation 5900 Folsom Blvd Sacramento, CA 95819 (916) 227-6975

1.1 Purpose

The primary purpose of the SRS is to define what requirements are needed for the proposed system along with assumptions, constraints, and dependencies for the design. Each large requirement has been broken down into smaller components to help describe exactly what each requirement must do.

The SRS will give the reader a better understanding of how ISD will handle each requirement during the design. SRS also serves as a baseline for designing and testing the proposed software. This document will basically serve as a checklist to verify that each requirement has been met.

1.2 Scope

The primary scope of the SRS is to describe how DRI views the requirements that were gathered by meeting with the sponsor. This document will also describe any assumptions, constraints, and dependencies that DRI foresees in designing the proposed software. Once approved by all parties in section 4, this document will become a baseline for the requirements needed for the proposed software. Once the SRS has become the baseline, changes to the document will only be made via a well-defined baseline change control process.

1.3 DefinitionS

Acronym	Definition
.NET	The Microsoft .NET platform for developing web sites and programs
ASP	Active Server Pages
DOT	Department of Transportation
DRI	Division of Research and Innovation
FSR	Feasibility Study Report
MM	Maintenance Manual
MySQL	An open source SQL server
PHP	Hypertext Preprocessor; General-purpose scripting language that can be used for web application development and can be embedded into HTML.
QA	Quality Assurance
SDD	Software Design Document: Document which describes in detail how project was implemented
SMM	Software Maintenance Manual: a technical maintenance manual that describes how to update and maintain a system after initial installation.
SPMP	Software Project Management Plan: a document that outlines the management of a software project.
SRS	Software Requirements Specification: a document which describes all the requirements related to the project.
STD	Software Testing Document: a document outlines the test procedures and strategies for the software
UM	User Manual

1.4 References

[SRS-4-20-02] – Buckley, R. (2002.) *Guide to Preparing the Software Requirements Specification Document*, Sacramento, CA

1.5 Overview of Contents of Document

Section 2 – General Description

The general description defines how the proposed system will be designed to and installed onto the sponsor’s current environment and describes the interfaces required to function with existing system. Section 2 will also contain use case models that will depict a user or external systems interaction with the proposed software. Assumptions, constraints and dependencies will also be listed and described in this section.

Section 3 – Specific Requirements

Specific Requirements will contain the detailed information about each of the use cases described in section 2. Here each use case will be outlined in a table that defines key information about it. The table also describes the pre and post conditions of the use case as well as the expected outcome of the use case. Additional information in this section pertains to the performance of the proposed system, any design constraints that are foreseen and the quality attributes that are part of the non-functional requirements.

Section 4 - Approvals

Approvals section is a place holder for the approving parties to indicate that they approve of the document in both context and content. Once this section is completed with signatures of all parties, the document becomes a baseline for the proposed system.

Appendix A – Data Dictionary

Appendix A includes the information about that data that the database will store. This is a data dictionary. These descriptions include the data types for each attribute of the tables in the system database that the proposed software will be using.

2 General Description

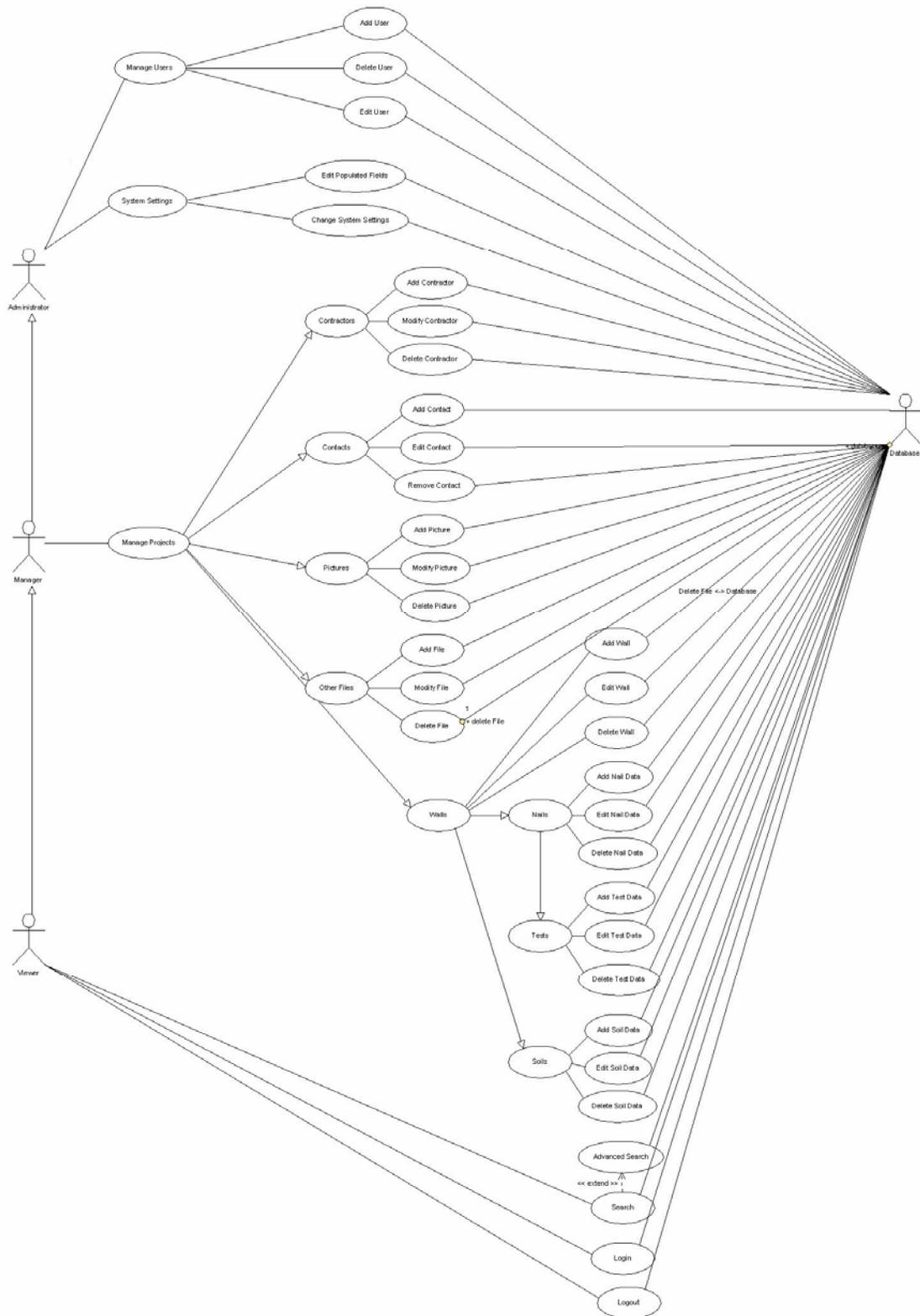
Within this section are the definitions for how CalNail will interact with its users as well as other external systems. Use case diagrams help define the different possible operations a user can perform and will be described in section 2.3. Details of each use case will be defined in section 3.1. Constraints on how the system will operate along with assumptions and dependencies required to make the system operational will also be defined. These will be found in section 2.4 and section 2.5 respectively.

2.1 Product Perspective

CalNail will be developed using web-based technologies PHP and MySQL. We will use a DRI server to house the database and application for the duration of the test deployment period. As a web application, this system will be available to all employees via intranet. As a web based application, there will be no need for additional software or hardware.

2.2 Use Case Diagram

Figure 2.2 of this section diagrams the actions that are available to the system's users and the interactions between the proposed system and other external systems. This diagram will give the overall description on how the users will use the proposed system.



Created with Poseidon for UML Community Edition. Not for Commercial Use.

2.3 User Characteristics

This will identify the use cases of each user in the system and their characteristics.

2.3.1 ADMINISTRATORS

Administrators are at the top of the user access rights. They alone have the power to remove users, edit/add/delete other users records, and perform database maintenance. Because of the potentially destructive nature of giving too much freedom to users Administrator accounts should be kept to a minimum.

- Add/Remove to Abbreviations and Symbols Table
- Add/Remove Users
- Change System Settings
- Database Administration
- Edit Remove Records
- Editing populated fields
- Searching by fields
- Submit binary and picture data
- Update Profiles
- Viewing Changed Records Log

2.3.2 MANAGERS

Managers are able to add to the database but limited to editing and deleting project data. They are also able to upload binary data.

- Add Records
- Edit Records
- Search by field types
- Upload pictures/data

2.3.3 VIEWERS

At the very bottom of the access chain viewers can only search and print data. They can also see various documents but that's all.

- Search by fields
- Print

2.4 General Constraints

Constraints limiting the ability to design the CalNail System include the following situations.

The following system resources must be available to the design they system:

- Server capable of running PHP Code
- Available database in MySQL

The user's system must have these resources available to use the designed software:

- A working World Wide Web browser that has the capabilities of sending data using web forms to use as the interface.
- A connection to the Caltrans Intranet

Other constraints that may limit the design team's ability:

- Time constraints play a big role in the ability to complete the project in a timely fashion. As the development team is primarily students school schedules will come as a priority to making a project deadline. If such events occur a new revised schedule will be issued to meet the needs of the students.

2.5 Assumptions and Dependencies

In the design of this project DRI is assuming that after the 18 month test period our client will have a capable web server with support for PHP and access to a MySQL database.

3 Specific Requirements

This section defines the use case for each case diagramed in section 2.2

3.1 Use Case Specifications

In the following subsections for 3.1 tables will describe the actions performed by each use case diagramed in section 2.3.

3.1.1 LOGIN

The following table describes the use case Login. It details the events that need to happen in order for the user to log into the system.

Use Case Name:	Login
Summary:	User will login by providing a valid username.
Basic Course of Events:	<ol style="list-style-type: none">1. Use case begins when user enters the system.2. System displays login screen.3. User inputs a valid username and password.4. System validates username and password.
Alternative Paths:	At any time, if the user cancels, the login screen closes.
Exception Paths:	In step 3, if the user inputs an invalid username or password, the user is prompted to re-enter username and password.
Trigger:	User requests to login.
Assumptions:	User has already signed up for a CalNail account
Preconditions:	User has a valid username and password.
Postconditions:	Welcome screen is presented to User
Author:	Jared Parnell
Date:	7/27/2005

Table 3-1-1 Login

3.1.2 LOGOUT

The following table describes the use case Login. It details the events that need to happen in order for the user to log out of the system.

Use Case Name:	Logout
Summary:	User exits the system
Basic Course of Events:	<ol style="list-style-type: none">1. Use case begins when user selects logout from the menu2. System displays main screen
Alternative Paths:	Once started there is no alternative
Exception Paths:	None
Trigger:	User requests to log out
Assumptions:	User has already signed in
Preconditions:	User is logged into the system
Postconditions:	Main screen is presented to the user
Author:	Jared Parnell
Date:	7/27/2005

Table 3-1-2 Logout

3.1.3 REGISTER

The following table describes the use case Register. It details the events that need to happen in order for the user to sign up to use the system.

Use Case Name:	Register
Summary:	User signs up for an account.
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects sign up from the menu 2. System displays registration form 3. User completely fills out form 4. User presses submit button
Alternative Paths:	User can cancel at any time to exit to main screen
Exception Paths:	If in step 3 the user doesn't fill out the form correctly a message appears along with the form in step 3.
Trigger:	User requests to sign up
Assumptions:	User doesn't have an existing account
Preconditions:	None
Postconditions:	User account is created
Author:	Jared Parnell
Date:	7/29/2005

Table 3-1-3 Register

3.1.4 SEARCH

The following table describes the use case Search. It details the events that need to happen in order for the user to start searching the database for information he/she would like.

Use Case Name:	Search
Summary:	User searches a record
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when the user selects Search from the main menu 2. System displays search page 3. User selects fields to search by 4. User presses submit button
Alternative Paths:	User can cancel at any time to exit to main screen
Exception Paths:	If in step 3 the user doesn't fill out the form correctly a message appears along with the form in step 3.
Trigger:	User requests to search
Assumptions:	User is logged in
Preconditions:	None
Postconditions:	Search results are displayed
Author:	Jared Parnell
Date:	7/29/2005

Table 3-1-4 Search

3.1.5 ADVANCED SEARCH

The following table describes the use case Advanced Search. It outlines the steps necessary to search via multiple criteria.

Use Case Name:	Advanced Search
Summary:	User searches database with specific request
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects advanced search from the search menu 2. System displays advanced search form 3. User fills out what he/she would like to search by 4. User presses submit button
Alternative Paths:	User can cancel at any time to exit to main screen
Exception Paths:	If in step 3 the user doesn't fill out the form correctly a message appears along with the form in step 3.
Trigger:	User requests advance search
Assumptions:	User is logged in
Preconditions:	None
Postconditions:	Search returns results page
Author:	Jared Parnell
Date:	7/29/2005

Table 3-1-5 Advanced Search

3.1.6 MANAGE PROJECTS

The following table describes the use case Manage Projects. It outlines the steps necessary to add, delete, and edit project information. From this case a user can add another project or edit/delete an existing one. This step is merely selection; other use cases actually do the work.

Use Case Name:	Manage Projects
Summary:	Allows user to add, delete, and edit projects
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Projects from the main menu 2. System displays projects in database 3. User selects a project to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the main screen at any time
Exception Paths:	If in step 3 the user doesn't select a project to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage projects
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	7/29/2005

Table 3-1-6 Manage Projects

3.1.7 ADD PROJECT

The following table describes the use case Add Project. It outlines the steps necessary to add a project to the database.

Use Case Name:	Add Project
Summary:	Allows user to add a new project to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add project from manage projects menu 2. System displays add project form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage projects menu
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add project
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Project is added and additional information screen is displayed.
Author:	Jared Parnell
Date:	7/29/2005

Table 3-1-7 Add Project

3.1.8 EDIT PROJECT

The following table describes the use case Edit Project. It outlines the steps necessary to make changes to the project information. The only information editable by this screen is the main project info..

Use Case Name:	Edit Project
Summary:	Allows a user to edit the main information about a project
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a project to edit from the manage projects menu 2. System displays the project information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the manage projects screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a project
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Project information is updated and edit screen is displayed for current project.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-8 Edit Project

3.1.9 DELETE PROJECT

The following table describes the use case Delete Project. It outlines the steps to remove a project from the database.

Use Case Name:	Delete Project
Summary:	Allows a user to remove a project from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a project to delete project from the manage projects menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the manage projects screen
Exception Paths:	None
Trigger:	User requests to delete a project
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Project and related information is stripped from the database.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-9 Delete Project

3.1.10 MANAGE CONTRACTORS

The following table describes the use case Manage Project Contractors. It outlines the steps necessary to add, delete, and edit project contractor information. From this case a user can add another project contractor or edit/delete an existing one.

Use Case Name:	Manage Contractors
Summary:	Allows user to add, delete, and edit project contractors
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Contractors from the Project Screen 2. System displays project's contractors in database 3. User selects a contractor to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the project information screen at anytime
Exception Paths:	If in step 3 the user doesn't select a contractor to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage project contractors
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-10 Manage Contractors

3.1.11 ADD CONTRACTOR

The following table describes the use case Add Contractor. It outlines the steps necessary to add a contractor to a project.

Use Case Name:	Add Contractor
Summary:	Allows user to add a new contractor to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add contractor from the project information screen 2. System displays add contractor form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage project information screen
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add contractor
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Contractor is added and project information screen is displayed.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-11 Add Contractor

3.1.12 EDIT CONTRACTOR

The following table describes the use case Edit Contractor. It outlines the steps necessary to make changes to the contractor information.

Use Case Name:	Edit Contractor
Summary:	Allows a user to edit the main information about a project
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a contractor to edit from the project information screen. 2. System displays the contractor information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the project information screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a contractor
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Contractor information is updated and edit screen is displayed for current contractor.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-12 Edit Contractor

3.1.13 DELETE CONTRACTOR

The following table describes the use case Delete Contractor. It outlines the steps to remove a contractor from a project.

Use Case Name:	Delete Contractor
Summary:	Allows a user to remove a project from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a project to delete project from the manage projects menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the project information screen
Exception Paths:	None
Trigger:	User requests to delete a contractor
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Contractor is removed from the project database
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-13 Delete Contractor

3.1.14 MANAGE CONTACTS

The following table describes the use case Manage Contacts. It outlines the steps necessary to add, delete, and edit project Contact information. From this case a user can add another contact or edit/delete an existing one.

Use Case Name:	Manage Contacts
Summary:	Allows user to add, delete, and edit project Contacts
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Contacts from the Project Screen 2. System displays project's Contacts in database 3. User selects a Contact to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the project information screen at anytime
Exception Paths:	If in step 3 the user doesn't select a Contact to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage project Contacts
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-14 Manage Contacts

3.1.15 ADD CONTACT

The following table describes the use case Add Contact. It outlines the steps necessary to add a Contact to a project.

Use Case Name:	Add Contact
Summary:	Allows user to add a new Contact to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add Contact from the project information screen 2. System displays add contactor form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage project information screen
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add Contact
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Contact is added and project information screen is displayed.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-15 Add Contact

3.1.16 EDIT CONTACT

The following table describes the use case Edit Contact. It outlines the steps necessary to make changes to the Contact information.

Use Case Name:	Edit Contact
Summary:	Allows a user to edit the main information about a project
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Contact to edit from the project information screen. 2. System displays the Contact information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the project information screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a Contact
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Contact information is updated and edit screen is displayed for current Contact.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-16 Edit Contact

3.1.17 DELETE CONTACT

The following table describes the use case Delete Contact. It outlines the steps to remove a Contact from a project.

Use Case Name:	Delete Contact
Summary:	Allows a user to remove a project from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a project to delete project from the manage projects menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the project information screen
Exception Paths:	None
Trigger:	User requests to delete a Contact
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Contact is removed from the project database
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-17 Delete Contact

3.1.18 MANAGE PICTURES

The following table describes the use case Manage Pictures. It outlines the steps necessary to add, delete, and edit project Picture. From this case a user can add another project Picture or edit/delete an existing one. This step is merely selection; other use cases actually do the work.

Use Case Name:	Manage Pictures
Summary:	Allows user to add, delete, and edit project Pictures
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Pictures from the Project Screen 2. System displays project's Pictures in database 3. User selects a Picture to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the project information screen at anytime
Exception Paths:	If in step 3 the user doesn't select a Picture to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage project Pictures
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-18 Manage Pictures

3.1.19 ADD PICTURE

The following table describes the use case Add Picture. It outlines the steps necessary to add a Picture to a project.

Use Case Name:	Add Picture
Summary:	Allows user to add a new Picture to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add Picture from the project information screen 2. System displays add Picture form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage project information screen
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add Picture
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Picture is added and project information screen is displayed.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-19 Add Picture

3.1.20 EDIT PICTURE

The following table describes the use case Edit Picture. It outlines the steps necessary to make changes to the Picture information.

Use Case Name:	Edit Picture
Summary:	Allows a user to edit the main information about a project
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Picture to edit from the project information screen. 2. System displays the Picture information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the project information screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a Picture
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Picture information is updated and edit screen is displayed for current Picture.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-20 Edit Picture

3.1.21 DELETE PICTURE

The following table describes the use case Delete Picture. It outlines the steps to remove a Picture from a project.

Use Case Name:	Delete Picture
Summary:	Allows a user to remove a project from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a project to delete project from the manage projects menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the project information screen
Exception Paths:	None
Trigger:	User requests to delete a Picture
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Picture is removed from the project database
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-21 Delete Picture

3.1.22 MANAGE FILES

The following table describes the use case Manage Files. It outlines the steps necessary to add, delete, and edit project File. From this case a user can add another project File or edit/delete an existing one. This step is merely selection; other use cases actually do the work.

Use Case Name:	Manage Files
Summary:	Allows user to add, delete, and edit project Files
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Files from the Project Screen 2. System displays project's Files in database 3. User selects a File to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the project information screen at anytime
Exception Paths:	If in step 3 the user doesn't select a File to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage project Files
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-22 Manage Project Files

3.1.23 ADD FILE

The following table describes the use case Add File. It outlines the steps necessary to add a File to a project.

Use Case Name:	Add File
Summary:	Allows user to add a new File to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add File from the project information screen 2. System displays add File form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage project information screen
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add File
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	File is added and project information screen is displayed.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-23 Add File

3.1.24 EDIT FILE

The following table describes the use case Edit File. It outlines the steps necessary to make changes to the File information.

Use Case Name:	Edit File
Summary:	Allows a user to edit the main information about a project
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a File to edit from the project information screen. 2. System displays the File information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the project information screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a File
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	File information is updated and edit screen is displayed for current File.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-24 Edit File

3.1.25 DELETE FILE

The following table describes the use case Delete File. It outlines the steps to remove a File from a project.

Use Case Name:	Delete File
Summary:	Allows a user to remove a project from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a project to delete project from the manage projects menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the project information screen
Exception Paths:	None
Trigger:	User requests to delete a File
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	File is removed from the project database
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-25 Delete File

3.1.26 MANAGE WALLS

The following table describes the use case Manage Walls. It outlines the steps necessary to add, delete, and edit Wall information. From this case a user can add another Wall or edit/delete an existing one. This step is merely selection; other use cases actually do the work.

Use Case Name:	Manage Walls
Summary:	Allows user to add, delete, and edit Walls
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Walls from the main menu 2. System displays Walls in database 3. User selects a Wall to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the main screen at any time
Exception Paths:	If in step 3 the user doesn't select a Wall to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage Walls
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-26 Manage Walls

3.1.27 ADD WALL

The following table describes the use case Add Wall. It outlines the steps necessary to add a Wall to the database.

Use Case Name:	Add Wall
Summary:	Allows user to add a new Wall to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add Wall from manage Walls menu 2. System displays add Wall form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage Walls menu
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add Wall
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Wall is added and additional information screen is displayed.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-27 Add Wall

3.1.28 EDIT WALL

The following table describes the use case Edit Wall. It outlines the steps necessary to make changes to the Wall information. The only information editable by this screen is the main Wall info..

Use Case Name:	Edit Wall
Summary:	Allows a user to edit the main information about a Wall
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Wall to edit from the manage Walls menu 2. System displays the Wall information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the manage Walls screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a Wall
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Wall information is updated and edit screen is displayed for current Wall.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-28 Edit Wall

3.1.29 DELETE WALL

The following table describes the use case Delete Wall. It outlines the steps to remove a Wall from the database.

Use Case Name:	Delete Wall
Summary:	Allows a user to remove a Wall from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Wall to delete Wall from the manage Walls menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the manage Walls screen
Exception Paths:	None
Trigger:	User requests to delete a Wall
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Wall and related information is stripped from the database.
Author:	Jared Parnell
Date:	8/1/2005

Table 3-1-29 Delete Wall

3.1.30 MANAGE NAILS

The following table describes the use case Manage Nails. It outlines the steps necessary to add, delete, and edit Nail information. From this case a user can add another Nail or edit/delete an existing one. This step is merely selection; other use cases actually do the work.

Use Case Name:	Manage Nails
Summary:	Allows user to add, delete, and edit Nails
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Nails from the main menu 2. System displays Nails in database 3. User selects a Nail to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the main screen at any time
Exception Paths:	If in step 3 the user doesn't select a Nail to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage Nails
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-30 Manage Nails

3.1.31 ADD NAIL

The following table describes the use case Add Nail. It outlines the steps necessary to add a Nail to the database.

Use Case Name:	Add Nail
Summary:	Allows user to add a new Nail to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add Nail from manage Nails menu 2. System displays add Nail form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage Nails menu
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add Nail
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Nail is added and additional information screen is displayed.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-31 Add Nail

3.1.32 EDIT NAIL

The following table describes the use case Edit Nail. It outlines the steps necessary to make changes to the Nail information. The only information editable by this screen is the main Nail info..

Use Case Name:	Edit Nail
Summary:	Allows a user to edit the main information about a Nail
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Nail to edit from the manage Nails menu 2. System displays the Nail information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the manage Nails screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a Nail
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Nail information is updated and edit screen is displayed for current Nail.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-32 Edit Nail

3.1.33 DELETE NAIL

The following table describes the use case Delete Nail. It outlines the steps to remove a Nail from the database.

Use Case Name:	Delete Nail
Summary:	Allows a user to remove a Nail from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Nail to delete Nail from the manage Nails menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the manage Nails screen
Exception Paths:	None
Trigger:	User requests to delete a Nail
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Nail and related information is stripped from the database.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-33 Delete Nail

3.1.34 MANAGE TESTS

The following table describes the use case Manage Tests. It outlines the steps necessary to add, delete, and edit Test information. From this case a user can add another Test or edit/delete an existing one. This step is merely selection; other use cases actually do the work.

Use Case Name:	Manage Tests
Summary:	Allows user to add, delete, and edit Tests
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Tests from the main menu 2. System displays Tests in database 3. User selects a Test to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the main screen at any time
Exception Paths:	If in step 3 the user doesn't select a Test to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage Tests
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-34 Manage Tests

3.1.35 ADD TEST

The following table describes the use case Add Test. It outlines the steps necessary to add a Test to the database.

Use Case Name:	Add Test
Summary:	Allows user to add a new Test to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add Test from manage Tests menu 2. System displays add Test form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage Tests menu
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add Test
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Test is added and additional information screen is displayed.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-35 Add Test

3.1.36 EDIT TEST

The following table describes the use case Edit Test. It outlines the steps necessary to make changes to the Test information. The only information editable by this screen is the main Test info..

Use Case Name:	Edit Test
Summary:	Allows a user to edit the main information about a Test
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Test to edit from the manage Tests menu 2. System displays the Test information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the manage Tests screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a Test
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Test information is updated and edit screen is displayed for current Test.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-36 Edit Test

3.1.37 DELETE TEST

The following table describes the use case Delete Test. It outlines the steps to remove a Test from the database.

Use Case Name:	Delete Test
Summary:	Allows a user to remove a Test from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Test to delete Test from the manage Tests menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the manage Tests screen
Exception Paths:	None
Trigger:	User requests to delete a Test
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Test and related information is stripped from the database.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-37 Delete Test

3.1.38 MANAGE SOILS

The following table describes the use case Manage Soils. It outlines the steps necessary to add, delete, and edit Soil information. From this case a user can add another Soil or edit/delete an existing one. This step is merely selection; other use cases actually do the work.

Use Case Name:	Manage Soils
Summary:	Allows user to add, delete, and edit Soils
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Soils from the main menu 2. System displays Soils in database 3. User selects a Soil to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the main screen at any time
Exception Paths:	If in step 3 the user doesn't select a Soil to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage Soils
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-38 Manage Soils

3.1.39 ADD SOIL

The following table describes the use case Add Soil. It outlines the steps necessary to add a Soil to the database.

Use Case Name:	Add Soil
Summary:	Allows user to add a new Soil to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add Soil from manage Soils menu 2. System displays add Soil form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to manage Soils menu
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add Soil
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Soil is added and additional information screen is displayed.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-39 Add Soil

3.1.40 EDIT SOIL

The following table describes the use case Edit Soil. It outlines the steps necessary to make changes to the Soil information. The only information editable by this screen is the main Soil info..

Use Case Name:	Edit Soil
Summary:	Allows a user to edit the main information about a Soil
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Soil to edit from the manage Soils menu 2. System displays the Soil information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the manage Soils screen
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a Soil
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Soil information is updated and edit screen is displayed for current Soil.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-40 Edit Soil

3.1.41 DELETE SOIL

The following table describes the use case Delete Soil. It outlines the steps to remove a Soil from the database.

Use Case Name:	Delete Soil
Summary:	Allows a user to remove a Soil from the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a Soil to delete Soil from the manage Soils menu 2. Confirmation screen is displayed 3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the manage Soils screen
Exception Paths:	None
Trigger:	User requests to delete a Soil
Assumptions:	User is logged in as an admin or manager
Preconditions:	None
Postconditions:	Soil and related information is stripped from the database.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-41 Delete Soil

3.1.42 MANAGE USERS

The following table describes the use case Manage Users. It outlines the steps necessary to add, delete, and edit User information. From this case a user can add another User or edit/delete an existing one. This step is merely selection; other use cases actually do the work.

Use Case Name:	Manage Users
Summary:	Allows user to add, delete, and edit Users
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects Manage Users from the main menu 2. System displays Users in database 3. User selects a User to control via radials or adds new one 4. User presses the action button he/she would like to perform.
Alternative Paths:	User can go to the main screen at any time
Exception Paths:	If in step 3 the user doesn't select a User to edit and presses the edit button it will return a message and go back to step 1
Trigger:	User requests to manage users
Assumptions:	User is logged in as an admin
Preconditions:	None
Postconditions:	Goes to the use case selected (add/delete/edit)
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-42 Manage Users

3.1.43 ADD USER

The following table describes the use case Add User. It outlines the steps necessary to add a User to the system.

Use Case Name:	Add User
Summary:	Allows user to add a new User to the database
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects add User from manage Users menu 2. System displays add User form 3. User fills out form appropriately 4. User presses the submit button.
Alternative Paths:	User can cancel back to Manage Users menu
Exception Paths:	If in step 3 the user doesn't fill out the form appropriately an error message is displayed and step 3 is repeated
Trigger:	User requests to add User
Assumptions:	User is logged in as an admin
Preconditions:	None
Postconditions:	User is added to the system and user list is displayed
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-43 Add User

3.1.44 EDIT USER

The following table describes the use case Edit User. It outlines the steps necessary to make changes to the User information such as access rights and passwords.

Use Case Name:	Edit User
Summary:	Allows a user to edit the main information about a User
Basic Course of Events:	<ol style="list-style-type: none"> 1. Use case begins when user selects a User to edit from the manage Users menu 2. System displays the User information form 3. User edits information in the form 4. User presses the submit button
Alternative Paths:	User can go cancel back to the Manage Users menu
Exception Paths:	If in step 3 the user erases required fields an error is displayed and step 2 is repeated.
Trigger:	User requests to edit a User
Assumptions:	User is logged in as an admin
Preconditions:	None
Postconditions:	User information is updated and edit screen is displayed for current User.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-44 Edit User

3.1.45 DELETE USER

The following table describes the use case Delete User. It outlines the steps to remove a User from the database.

Use Case Name:	Delete User
Summary:	Allows a user to remove a User from the database
Basic Course of Events:	<ol style="list-style-type: none">1. Use case begins when user selects a User to delete User from the manage Users menu2. Confirmation screen is displayed3. User presses the confirmation button
Alternative Paths:	User can go cancel back to the manage Users screen
Exception Paths:	None
Trigger:	User requests to delete a User
Assumptions:	User is logged in as an admin
Preconditions:	None
Postconditions:	User information is stripped from the database.
Author:	Jared Parnell
Date:	8/2/2005

Table 3-1-45 Delete User

APPENDIX D:
CALNAIL
INSTALLATION OF CALNAIL

Calnail Installation:

Prerequisites:

- Windows or Linux OS
- Web Server capable of using PHP 5 (Apache/IIS)
- PHP 5 or better
- MySQL 4 or better
- PhpMyAdmin or MySQL Administrator

Downloads:

MySQL: <http://dev.mysql.com/downloads/>

- Choose the Windows x86 edition for easier setup

PHP5: <http://us3.php.net/downloads.php>

- Choose the 8Mb zip file (not the self installer)

PHPMyAdmin: http://www.phpmyadmin.net/home_page/downloads.php

- Choose the windows zip file

MySQL Administrator: <http://dev.mysql.com/downloads/administrator/1.1.html>

- Choose the windows x86 installer

Required Files:

- Calnail.zip
- PHP5
- **Recommended:**
 - o MySQL Administrator
- **Alternative:**
 - o PHPMyAdmin

Installation:

Web service basic installation will not be covered in this manual. Its also recommended you use “MySQL Adminsitrator” instead of phpmyadmin

If you need to install MySQL or PHP5 please use the following documents:

- MySQL Install.doc
- PHP Install.doc

Installation Procedure:

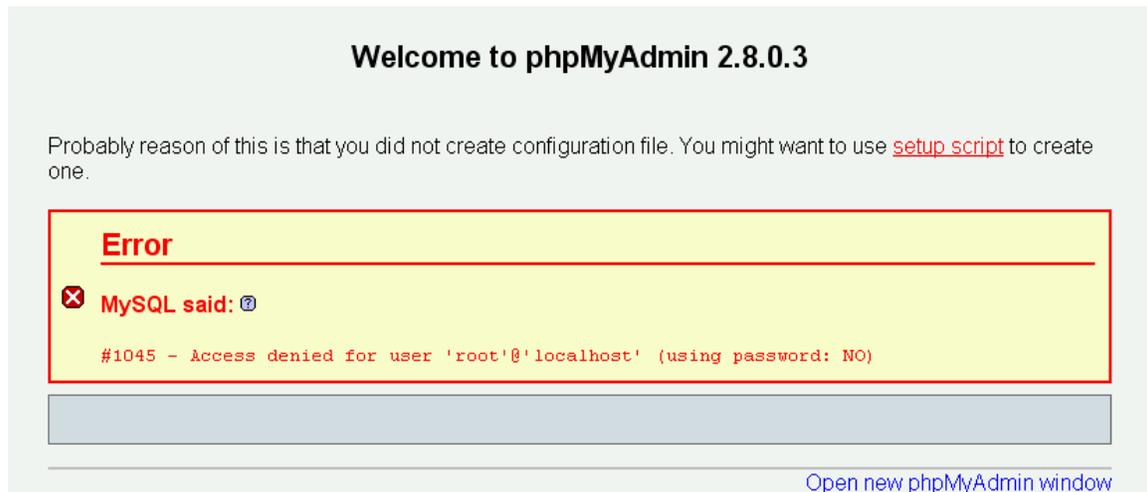
- 1) Download and use one of the two MySQL administrative programs. This installation recommends using MySQL Administrator.
- 2) Install the MySQL admin tool
- 3) Setup the Calnail database for use in calnail
- 4) Unzip the calnail.zip file into a web-folder
- 5) Edit the include_database.php file to reflect the database and password in your system.

MySQL Administrator: (Step 2)

- Download the setup file from the location described above
- Run the setup file
- Use all the defaults then finish
- Open the application in the MySQL programs folder
- Select for location “localhost”
- Use “root” as the user
- Use your written down password for root when you setup MySQL
- No other parameters need to be filled
- Finish the login process if successful you should get a administration page

PhpMyAdmin: (If using MySQL Administrator skip this section!) (Step 2)

- Open zip file
- Copy contents to a directory of your choosing
- Create a new folder called “config” and edit the security permission to make it read and writable by Everyone
- Open the webpage and click the “Setup Script” link



- Now click Add Servers button on the following page
- Enter the information for the server
- Click the save button
- Move the file created in the config folder to the main folder and reload the phpmyadmin webpage
- If correct information is present the phpmyadmin page will show up

Welcome

i You want to configure phpMyAdmin using web interface. Please note that this only allows basic setup, please read [documentation](#) to see full description of all configuration directives.

⚠ Not secure connection

You are not using secure connection, all data (including sensitive, like passwords) are transferred unencrypted!

Available global actions (please note that these will delete any changes you could have done above):

Servers					
<input type="button" value="Add"/>					
Layout					
<input type="button" value="Left frame"/>	<input type="button" value="Tabs"/>	<input type="button" value="Icons"/>	<input type="button" value="Browsing"/>	<input type="button" value="Editing"/>	<input type="button" value="Query window"/>
Features					
<input type="button" value="Upload/Download"/>	<input type="button" value="Security"/>	<input type="button" value="MySQL manual"/>	<input type="button" value="Charsets"/>	<input type="button" value="Extensions"/>	<input type="button" value="MIME/Relation/History"/>
Configuration					
<input type="button" value="Overview"/>	<input type="button" value="Display"/>	<input type="button" value="Download"/>	<input type="button" value="Save"/>	<input type="button" value="Load"/>	<input type="button" value="Clear"/>
Other actions					
<input type="button" value="Check for latest version"/>	<input type="button" value="Go to homepage"/>	<input type="button" value="Donate to phpMyAdmin"/>			

Setup Calnail Database (Step 3)

For larger SQL files use MySQL Administrator (skip down)

phpMyAdmin way:

Locate the calnail.sql file

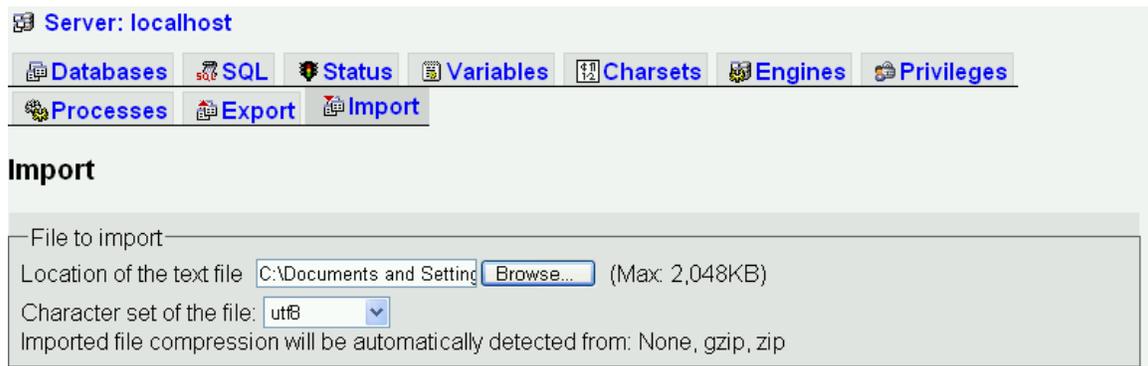
Create a new database named “calnail”

Open the phpMyAdmin webpage and click the Import link near the bottom of the page



Select the zipped Sql file and press the “Go” button

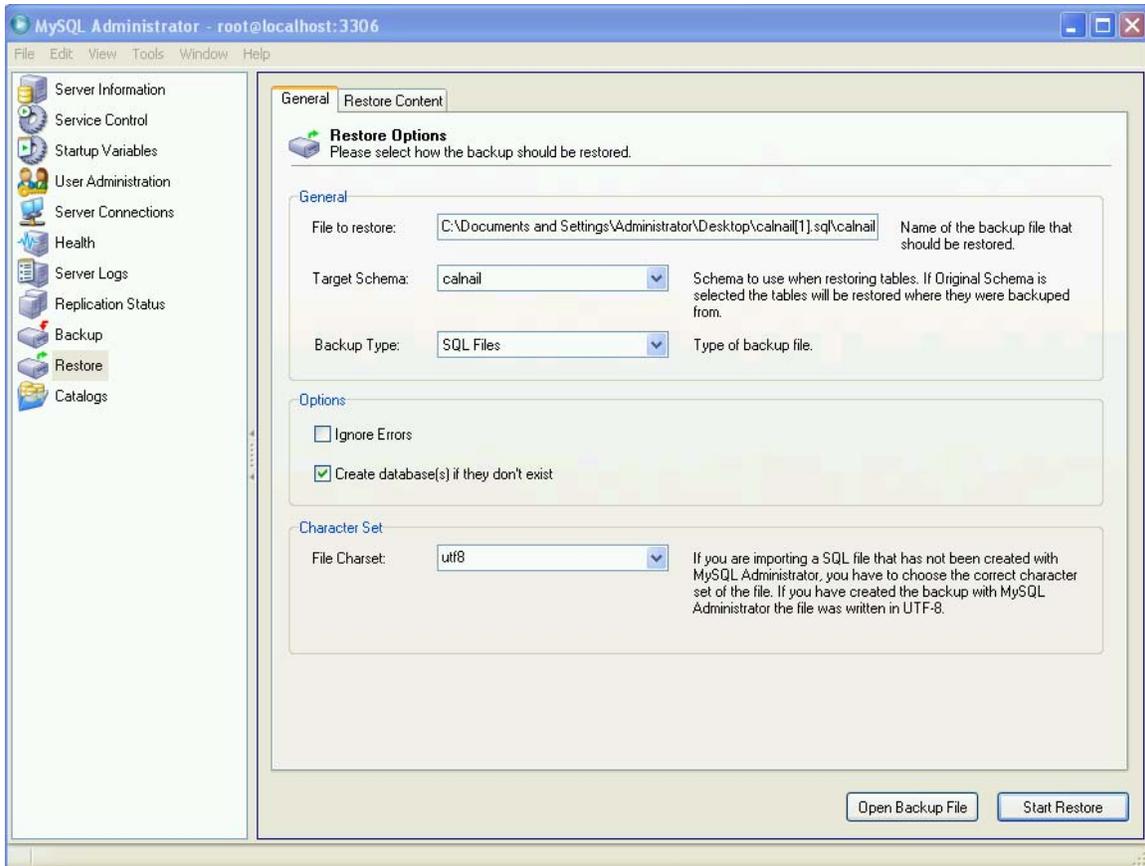
If you have problems ensure you have the temp directory have all permissions by Everyone and that the make file size isn't breached.



For extremely Large files its recommended you use MySQL Administrator

MySQL Administrator Way: (Step 3)

- Open the administrator client select the SQL database file
- Select Create database if they don't exist
- Select target schema "calnail"
- Start Restore and wait patiently for file to process
- Note: using the backup function you can effectively backup and restore your database from a prior period.



Setting up Calnail: (Step 4 and 5)

- Unzip the folder to a place in the web service directory
 - IE: c:\inetpub\wwwroot\calnail
 - Open the file include_database.php
 - Edit the file with a Text editor
 - Scroll down until you reach the line
//Setup the connection
- ```
$this->host = "localhost";
$this->db = "calnail";
$this->user = "root";
$this->pass = "";
$this->dbtype = $GLOBALS["DB_TYPE_MYSQL"];
```
- Edit the password to match your MySQL installation and save
  - Calnail should now be setup

# APPENDIX E: CALNAIL

INSTALLATION OF MYSQL DATABASE FOR CALNAIL

Open the setup file and follow the on screen instructions”



Choose “typical” installation:



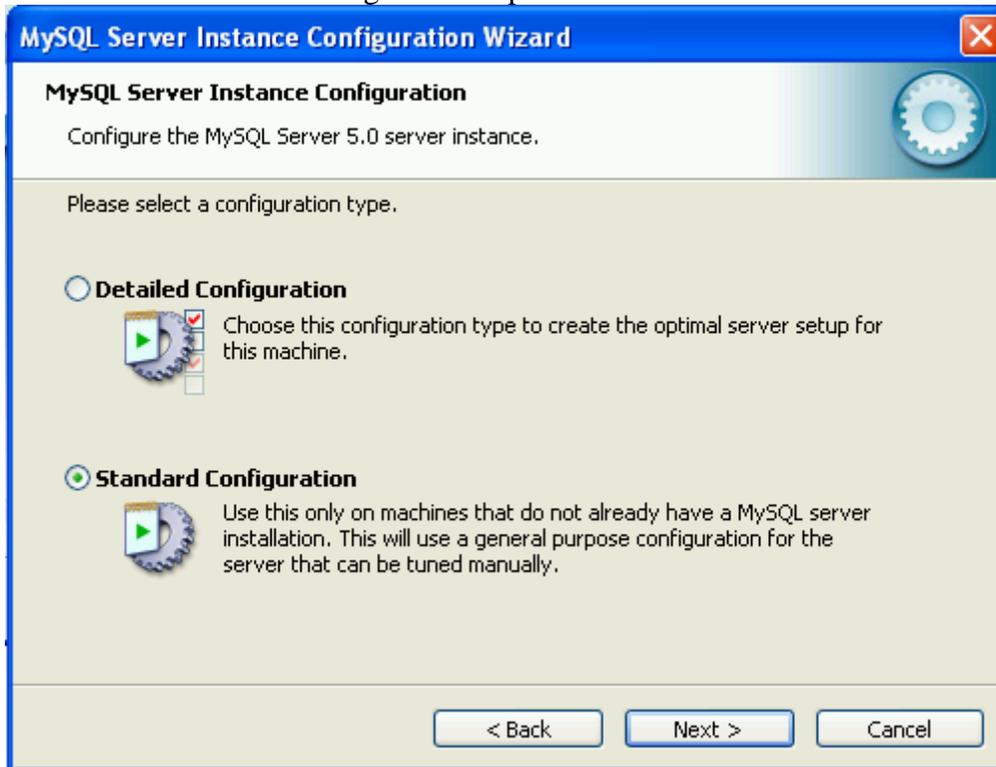
Choose the “Skip Sign-Up” option:



Choose the “Configure the MySQL Server now” option:



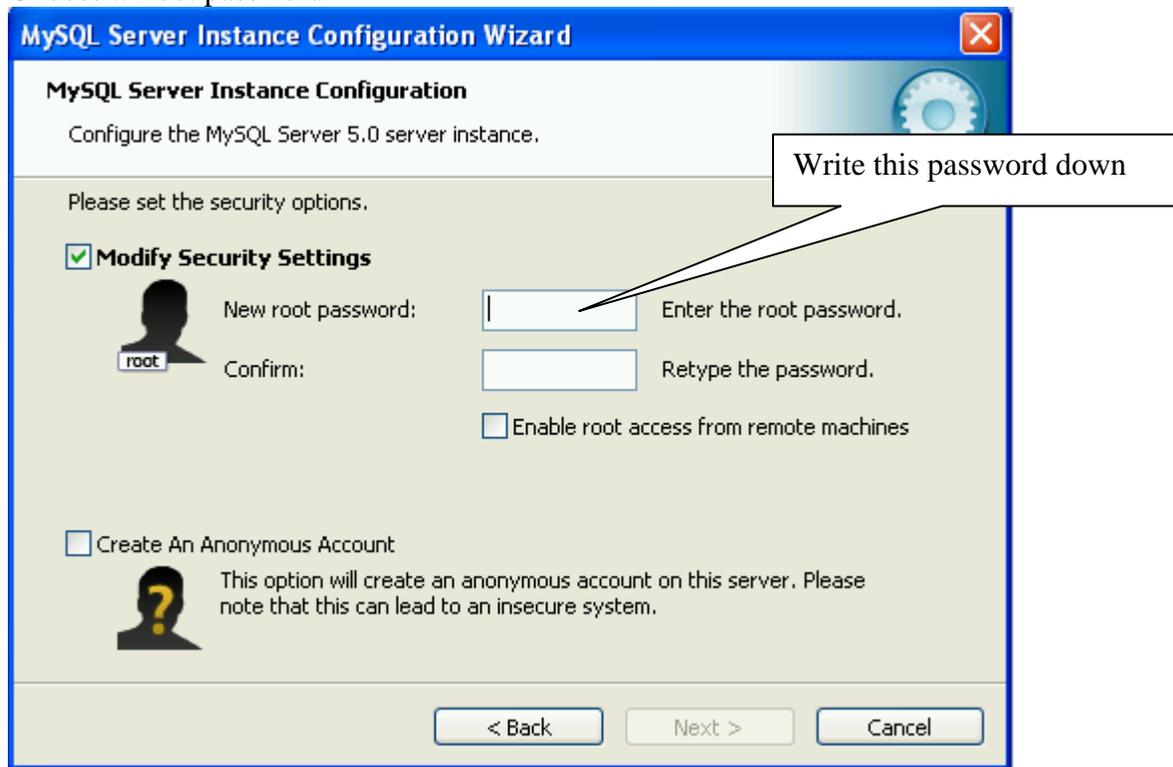
Choose the “Standard Configuration” option:



Standard Options



Choose a Root password



Continue and write settings to finish

**IMPORTANT:**

Calnail uses large BLOB sizes. This means the maximum allowable packet must be over 1MB.

To correct this issue a my.ini file is provided for you. Drop this file into the c:\programs files\mysql\ folder where the current my.ini files resides. Then restart the server.

# APPENDIX F: CALNAIL

INSTALLATION OF PHP FOR CALNAIL

PHP Install Documentation:

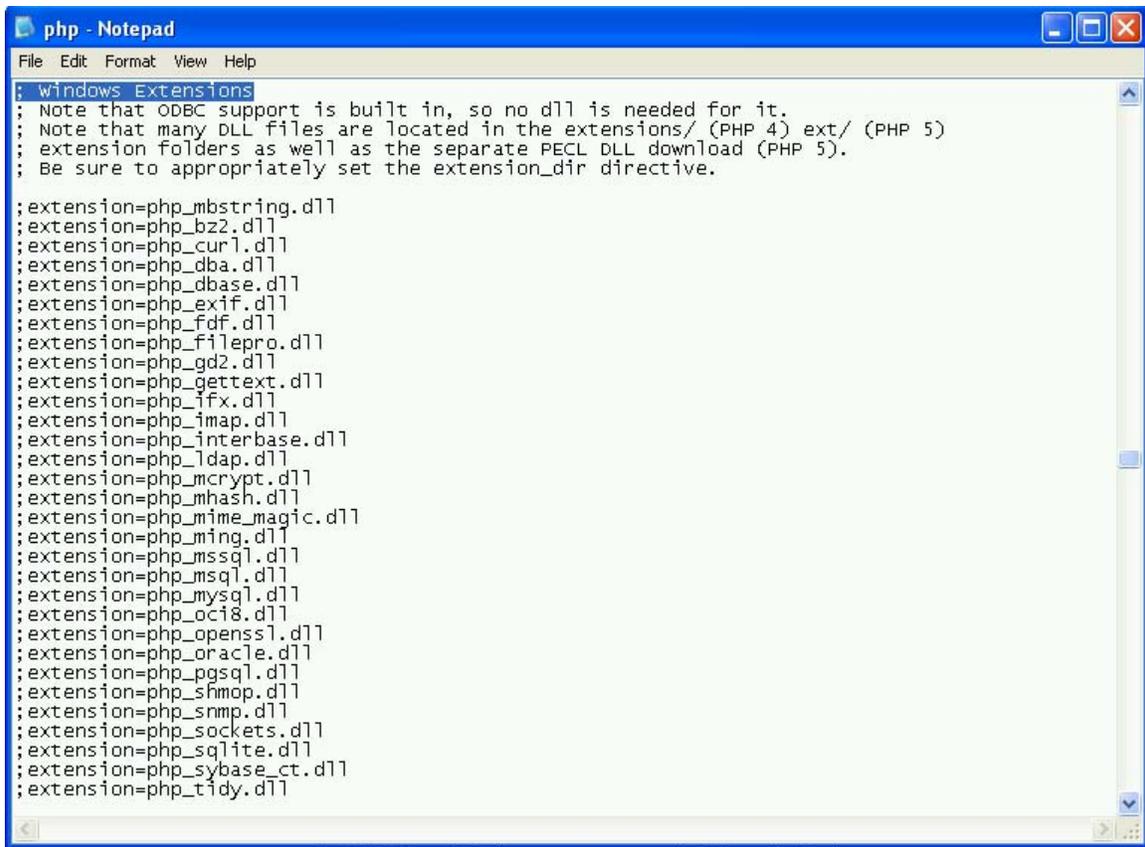
Requires you have a working Web server!

Right click and extract the Php Zip file to C:\PHP or a folder of your choice  
Edit the directory Security so Everyone and Read and Write to the directory



The folder will contain Installation instructions to install PHP in a text format. This guide will show how to install PHP with IIS.

- Rename php.ini-recommended to php.ini
- Edit the php.ini file with a Text Editor
- Set doc\_root = c:\inetpub\wwwroot
- Set extension\_dir = c:\php\ext
- Set cgi.force\_redirect = 0
- Search for the phrase "Windows Extensions"
- Uncomment the mbstring, gd2, and mysql extensions by removing the ";" in front (shown in the picture below)

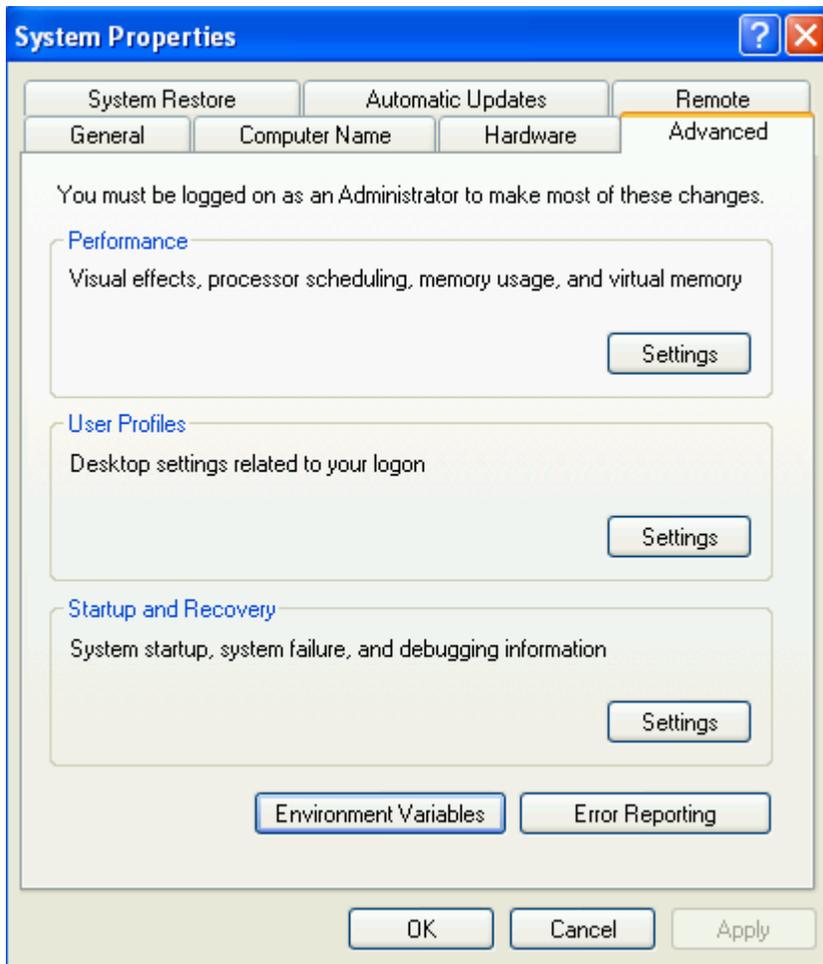


```
File Edit Format View Help
: windows Extensions
: Note that ODBC support is built in, so no dll is needed for it.
: Note that many DLL files are located in the extensions/ (PHP 4) ext/ (PHP 5)
: extension folders as well as the separate PECL DLL download (PHP 5).
: Be sure to appropriately set the extension_dir directive.

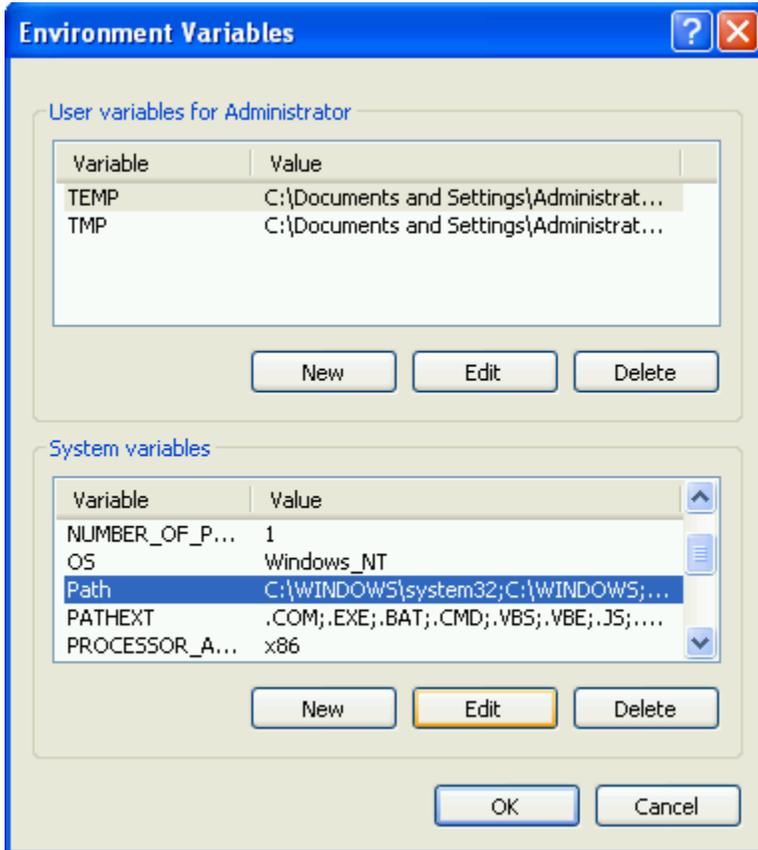
: extension=php_mbstring.dll
: extension=php_bz2.dll
: extension=php_curl.dll
: extension=php_dba.dll
: extension=php_dbase.dll
: extension=php_exif.dll
: extension=php_fdf.dll
: extension=php_filepro.dll
: extension=php_gd2.dll
: extension=php_gettext.dll
: extension=php_ifx.dll
: extension=php_imap.dll
: extension=php_interbase.dll
: extension=php_ldap.dll
: extension=php_mcrypt.dll
: extension=php_mhash.dll
: extension=php_mime_magic.dll
: extension=php_ming.dll
: extension=php_mssql.dll
: extension=php_mysql.dll
: extension=php_mysql.dll
: extension=php_oci8.dll
: extension=php_openssl.dll
: extension=php_oracle.dll
: extension=php_pgsql.dll
: extension=php_sfmop.dll
: extension=php_snmp.dll
: extension=php_sockets.dll
: extension=php_sqlite.dll
: extension=php_sybase_ct.dll
: extension=php_tidy.dll
```

Add the PHP directory to the PATH

- Goto the System Properties in the control panel
- Click the advanced tab
- Click Environment variables

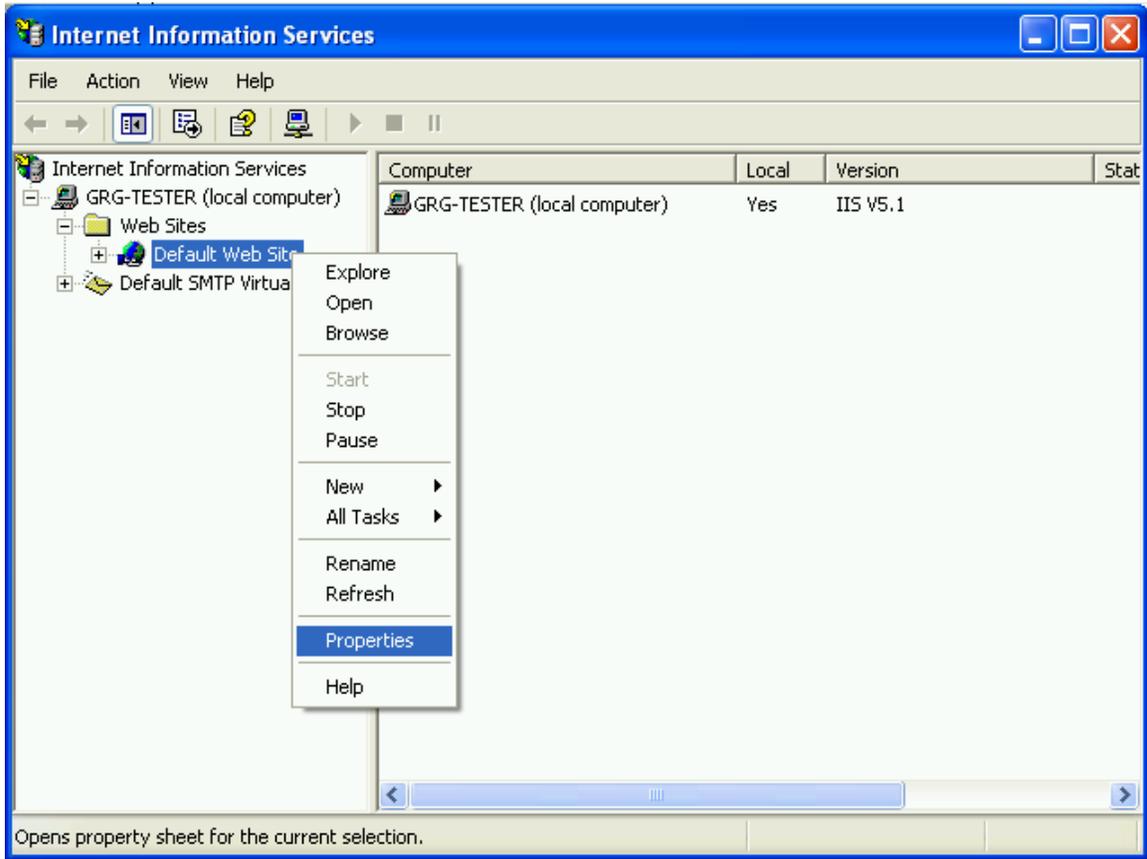


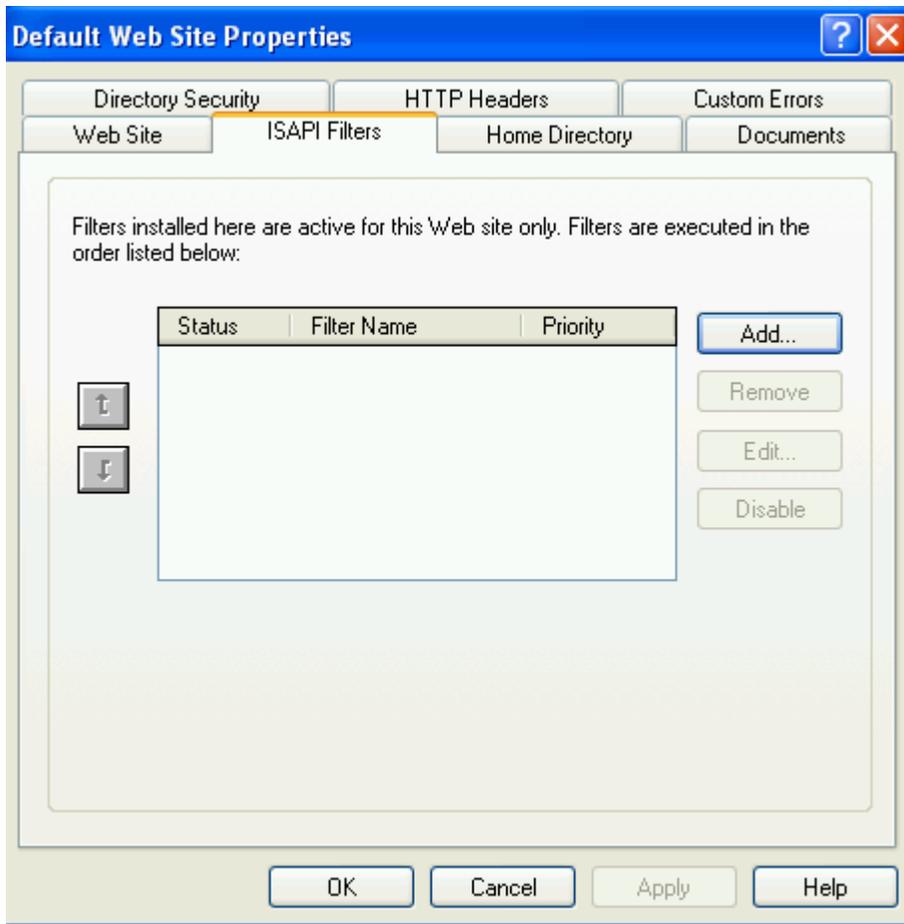
- Use the scroll bar to find “Path” and Edit
- Add the directory to the end of the path “;C:\PHP”
- Hit ok and save



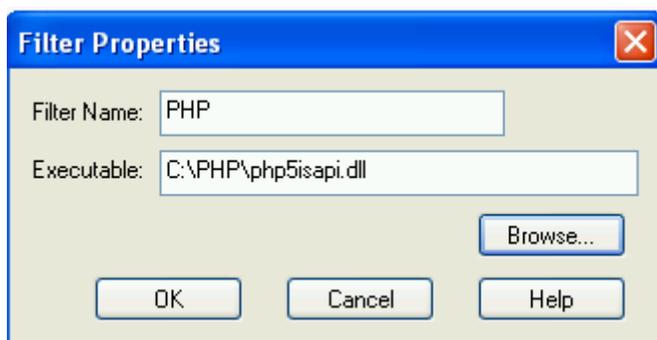
Open the Information Services Manager

- Right click the web sit you wish to add php extentions to and click properties
- Click the ISAPI filters tab

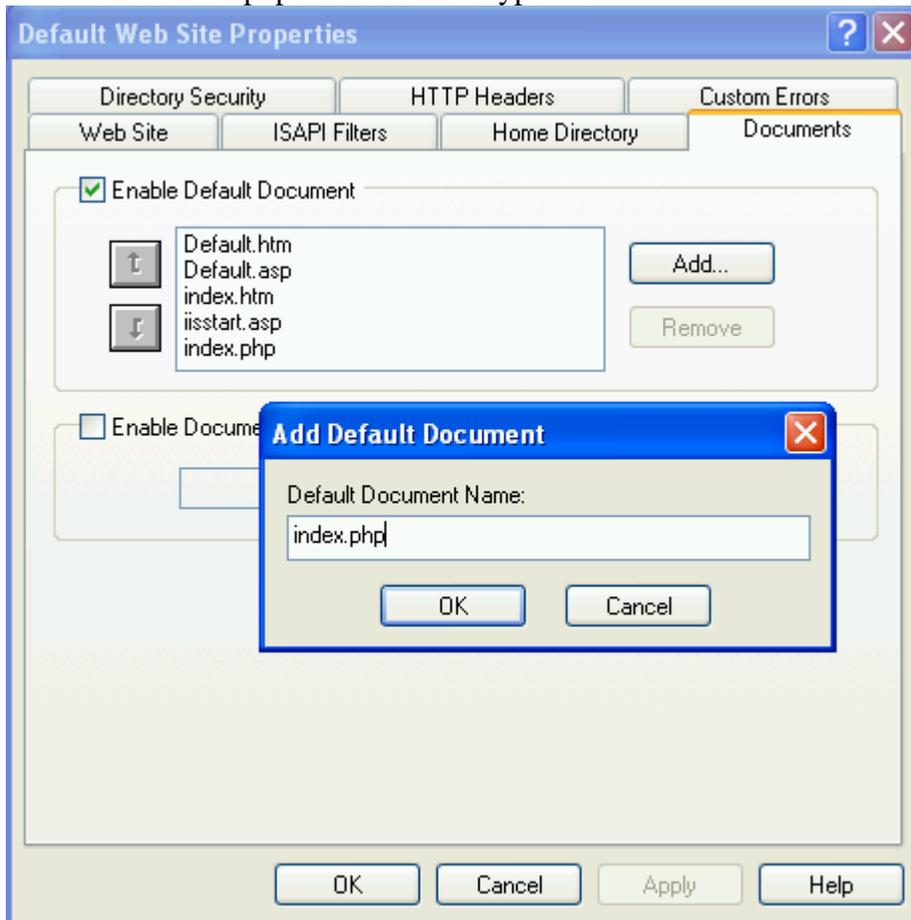




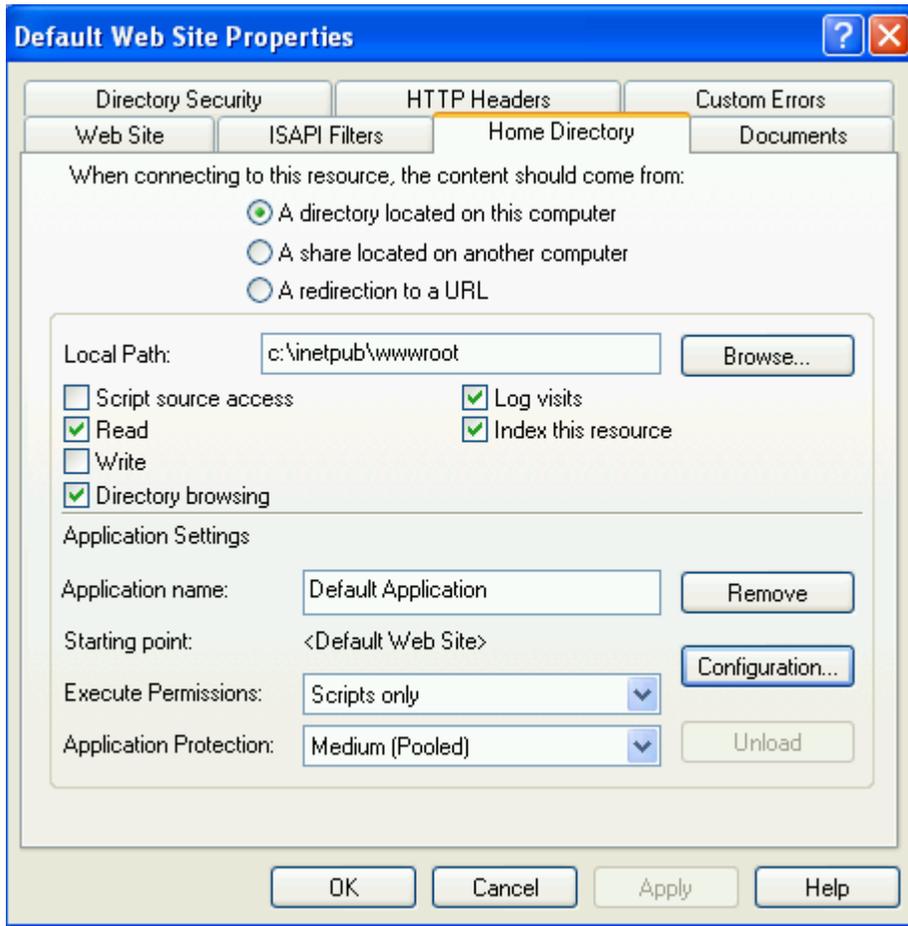
- Label the ISAPI filter "PHP"
- Browse for the php5isapi.dll in the C:\PHP directory
- Save the filter



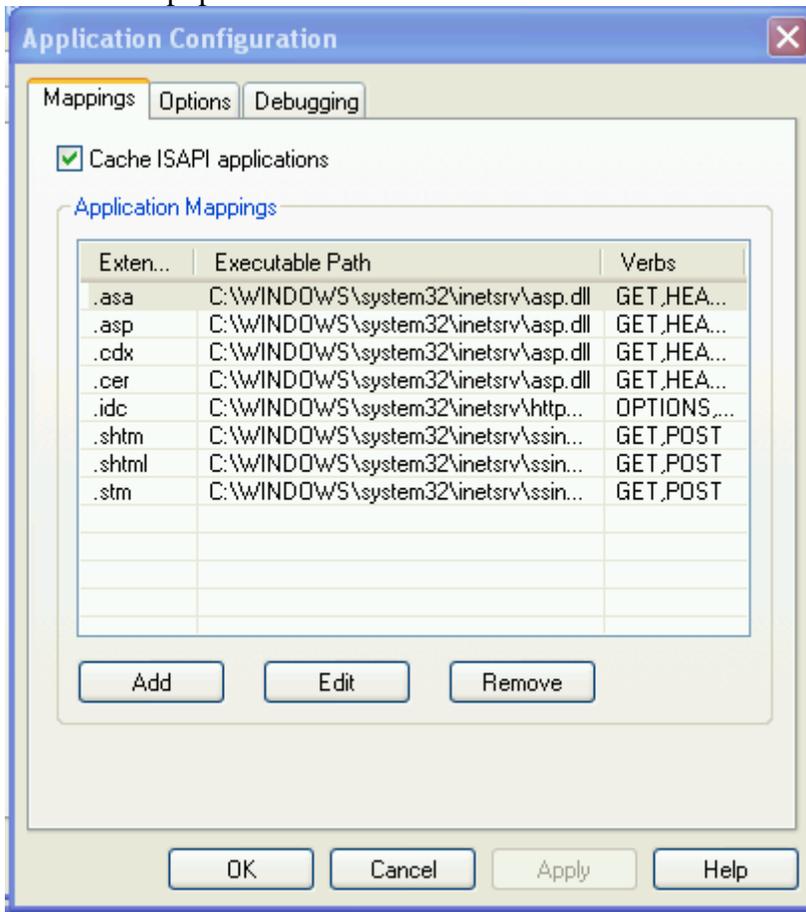
- Goto the Documents tab
- Add index.php as a document type



- Goto the Home Directory tab
- Click Configuration



- Click Configuration
- Add an extension
- Input the path to the php-cgi.exe
- Use .php as the extension and save



Restart the server and your finished

APPENDIX G:  
CALNAIL  
USER'S GUIDE

## Calnail Users Guide:

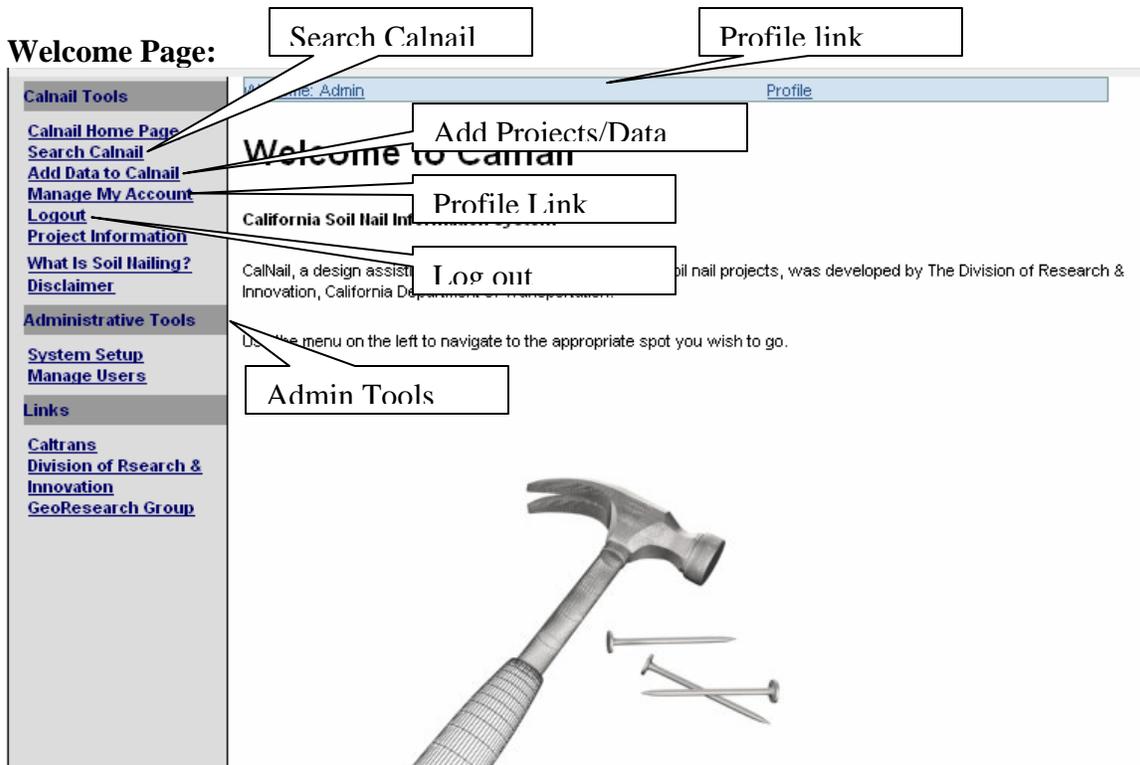
### Start Page:

The screenshot shows the Calnail website interface. On the left is a navigation menu with sections: 'Calnail Tools' (containing links for Home Page, Search Calnail, Logout, Project Information, What Is Soil Nailing?, and Disclaimer), and 'Links' (containing links for Caltrans, Division of Research & Innovation, and GeoResearch Group). The main content area features a search bar at the top right with the text 'Welcome: Please sign in to' and a callout 'Search Calnail (only valid users can)'. Below the search bar is a 'Log out button' callout. The central text reads: 'Callnail, a design assist... histories of soil nail pr... The Division of Researc... Department of Transportaon.' A callout 'Project Information' points to this text. Below this is a registration notice: 'To use the Callnail, you need to register. If you are already a registered user, please log in below.' A callout 'Login here' points to the 'LOG IN' button. Below that is another notice: 'If you are new to this site please sign up to search calnail.' A callout 'New Users click here' points to the 'SIGN UP' button. At the bottom, there are links for 'What is Soil Nailing Technology?' and 'Project Information'. On the right side of the page, there is a graphic showing soil nail installation and a photograph of a construction site.

When a user logs into calnail they will first see this login page. If you are a new user you need to sign up to utilize any of calnails functions. Use the signup button to get an account.

### New Accounts:

New accounts MUST have a valid Caltrans email. Email outside of Caltrans will not properly sent through (SMPT limitations). Once you fill out the form an email will be sent to you. Clicking on the link will validate your account and you will be free to use the search features only. If you need more privileges an administrator will be needed to grant you the permissions you require.



**Search Calnail:**

You can search calnail with this link using 4 types of search criteria. You can issue a search by Project, Soil, Wall, and Nail information. These searches in turn will hand a results page (which is read only) of the following requested results if any.

**Add Projects/Data:**

Managers or Admins can add to the database with this link. You can also edit projects and data from this page.

**Log out:**

Close out your connection when through browsing calnail here.

**Profile Link:**

This allows a validated user to edit their profile and change their personal information or change their password if entered.

**Admin tools:**

System Setup is a list manager that gives you as an admin the ability to edit pull down menus by adding/editing/removing items from the database.

Manage Users is a simple user editor that allows the admin to grant permissions and validate users (if need be).

# Search Calnail

Use the menu to add,edit,delete from the lists

## Search By Type

| List of Searches |                     |
|------------------|---------------------|
| Search           | Project Information |
| Search           | Wall Information    |
| Search           | Nail Information    |
| Search           | Soil Information    |

Filling out any of the fields forces a search by that criteria. Leaving it blank is the same as a wildcard (grab all) search.

## Search by Projects

| Project Information                   |                                                                                                                                         |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Project Name                          | <input type="text"/>                                                                                                                    |
| District                              | Any <input type="button" value="v"/>                                                                                                    |
| County                                | Any <input type="button" value="v"/>                                                                                                    |
| City                                  | <input type="text"/>                                                                                                                    |
| Route                                 | <input type="text"/> eg:route 80                                                                                                        |
| (Post Mile /Mile Post)                | <input type="text"/> To <input type="text"/> Any <input type="button" value="v"/> eg: [R] 90 / [R] 120                                  |
| Bridge                                | <input type="text"/> Any <input type="button" value="v"/> <input type="text"/> <input type="text"/> eg: 34-0012 L                       |
| EA Number                             | <input type="text"/>                                                                                                                    |
| Project Completion                    | <input type="text"/> To <input type="text"/> eg: 2004-12-31 (YYYY-MM-DD)                                                                |
| Location Information                  |                                                                                                                                         |
| Location Description                  | <input type="text"/>                                                                                                                    |
| LAT / LONG Coordinates                |                                                                                                                                         |
| LAT                                   | Min: <input type="text"/> Max: <input type="text"/>                                                                                     |
| LONG                                  | Min: <input type="text"/> Max: <input type="text"/>                                                                                     |
| Datum                                 | Any <input type="button" value="v"/>                                                                                                    |
| Northing Easting Coordinates          |                                                                                                                                         |
| Northing                              | Min: <input type="text"/> Min: <input type="text"/>                                                                                     |
| Easting                               | Min: <input type="text"/> Min: <input type="text"/>                                                                                     |
| LCRS                                  | Any <input type="button" value="v"/>                                                                                                    |
| Station + Offset [L/R]                |                                                                                                                                         |
| Station / Offset                      | Station <input type="text"/> + <input type="text"/> To <input type="text"/> + <input type="text"/> Any <input type="button" value="v"/> |
| Reference Line                        | <input type="text"/>                                                                                                                    |
| Elevation                             |                                                                                                                                         |
| Vertical Datum                        | Any <input type="button" value="v"/>                                                                                                    |
| Elevation                             | Min: <input type="text"/> Max: <input type="text"/> Feet <input type="button" value="v"/>                                               |
| Survey Method                         |                                                                                                                                         |
| Survey Type                           | Any <input type="button" value="v"/>                                                                                                    |
| Accuracy                              | Any <input type="button" value="v"/>                                                                                                    |
| <input type="button" value="Submit"/> |                                                                                                                                         |

Searching is straightforward, if you enter criteria to search for it will filter based on that input. No input is the same as searching ALL records.

## Add Projects/Data: Project Management

Use the menu to add, edit, or delete projects

| Project                  | District                | Route | County         | Date       | Edit                 | Delete                 |
|--------------------------|-------------------------|-------|----------------|------------|----------------------|------------------------|
| Bayshore Mall            | Eureka                  | 101   | Humboldt       | 1998-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Beale Street             | Oakland                 | 80    | San Francisco  | 1999-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Blue Hills               | Oakland                 | 80    | San Francisco  | 1991-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Camino Ruiz              | San Diego               | 15    | San Diego      | 2006-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Cemetery Project         | Eureka                  | 101   | Humboldt       | 2006-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Clairemont Mesa          | San Diego               | 15    | San Diego      | 2006-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Contra Costa Route 680   | Oakland                 | 680   | Contra Costa   | 1994-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Contra Costa Route 80    | Oakland                 | 80    | Contra Costa   | 1994-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Diamond Bar              | San Bernardino          | 57    | Los Angeles    | 2001-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| El Cerrito               | Oakland                 | 80    | Contra Costa   | 1992-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Fletcher Parkway         | San Diego               | 125   | San Diego      | 1999-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Sacramento Route 50      | Marysville/North Region | 50    | Sacramento     | 1990-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Hot Wash                 | San Bernardino          | 15    | San Bernardino | 1999-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| LA Route 405             | Los Angeles             | 405   | Los Angeles    | 2000-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| LA Route 710             | Los Angeles             | 710   | Los Angeles    | 1997-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Lake Tahoe Route 89      | Marysville/North Region | 89    | Placer         | 1994-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| San Bernardino Route 210 | San Bernardino          | 210   | San Bernardino | 2001-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| San Bernardino Route 71  | San Bernardino          | 71    | San Bernardino | 1995-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Whittier Undercrossing   | Los Angeles             | 710   | Los Angeles    | 1997-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Wightman Overcrossing    | San Diego               | 15    | San Diego      | 1998-01-01 | <a href="#">Edit</a> | <a href="#">Delete</a> |
| Turner Hill              |                         | 101   | Monterey       | 2006-05-12 | <a href="#">Edit</a> | <a href="#">Delete</a> |

Edit projects

Remove Project

Add Project

Add New Project

Only Managers and Admins can use these pages to add, edit, delete projects from the database. The next section will show how to interact with add/edit pages.

Editing pages will show related fields, which are associated with the project, soil, nail, or wall requested for edit.

Deleting pages will require confirmation like the screen shown below.

## Adding Data:

Fill in the blanks to add basic project data  
All **RED** fields are required

The screenshot shows a web form titled "Add Project Form" with the following fields:

- Project Name:** A text input field containing "test", highlighted with a red border and a callout box labeled "Required Field".
- District:** A dropdown menu showing "1 - Eureka".
- County:** A dropdown menu showing "ALA - Alameda".
- City:** An empty text input field.
- Route:** An empty text input field with a hint "eg: 80".
- (Post Mile /Kilo Post):** A form with two sets of input fields (number, dash, number, slash, number, dash, number) and a "KiloPost" dropdown menu. A hint "eg: [R] 90 / [R] 120" is shown.
- Bridge:** A form with two input fields, a dropdown menu, and another input field. A hint "eg: 34-0012 L" is shown.
- EA Number:** An empty text input field.
- Project Completion:** An empty text input field with a hint "eg: 2004-12-31 (YYYY-MM-DD)".
- Comments:** A large text area.

A "Submit" button is located at the bottom left of the form. A "Confirmation Screen" dialog box is overlaid on the form, titled "Microsoft Internet Explorer" and containing the text "Are you SURE you want to add this record?" with "OK" and "Cancel" buttons.

When adding information there are fields, which are required, to successfully add any information to the database. When a successful entry of this is entered and submitted a confirmation screen will appear to ensure you are complete with your filing.

Data type checking is preformed before confirmation to ensure any non valid datatypes are not entered.

After adding information some screens will move to an Edit portion which allows the edition of more "related" data to the project, wall, nail, or soil.

## Edit Data/Information:

**Edit Project Form**

Project Name: Turner Hill

District: 5 - San Luis Obispo

County: MTY - Monterey

City: Monterey

Route: 101 eg: I-80

(Post Mile /Kilo Post): R - 20.2 / R - 23.1 Kilometers eg: [R] 90 / [R] 120

Bridge: - eg: 34-0012 L

EA Number: 123456

Project Completion: 2006-05-12 eg: 2004-12-31 (YYYY-MM-DD)

Comments: This is a test project.

Submit Reset

**Locations** Contractors Contacts Files Walls

| Location     | Coordinates | NE | Station | Line |
|--------------|-------------|----|---------|------|
| Add Location |             |    |         |      |

Related data will be stored below the main data above required to start a project "file". Clicking a tab will switch to that data.

When editing a project or other large associated collection of information like wall's or soils. Related data sets will be shown on the bottom in the form of a tabbed menu. Adding data to these related fields is the same as adding to the main project despite their different requirements.

Using the edit button will allow you to parse down to other related subsets. Wall data which contains nail and soils information. Editing the soil or nail information in turn shows their subsets.

## Admin Tools: (System Setup)

### Lists of Named Items

| List of figures |                     |
|-----------------|---------------------|
| Edit List       | Accuracy Listing    |
| Edit List       | Bar Types Listing   |
| Edit List       | Patern Listing      |
| Edit List       | Software Listing    |
| Edit List       | Soil Types Listing  |
| Edit List       | Survey Type Listing |

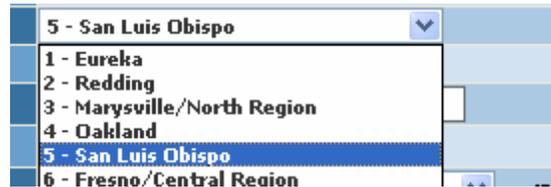
### Code Lists

| List of codes |                        |
|---------------|------------------------|
| Edit List     | Datum Listing          |
| Edit List     | Vertical Datum Listing |

### Specialized Lists

| Specialized lists |                  |
|-------------------|------------------|
| Edit List         | District Listing |
| Edit List         | County Listing   |

Editing these lists effects the drop down menus. Such as this drop down menu which is the district listing. By editing these menus you as an administrator can remove, add, change the names in these drop down menus.



Each menu is a simple 1 or 2 field edit screen with their own required fields and listing mechanisms.

## Admin Tools: (User management) User Management

Use the menu to add, edit, or delete users

| User    | Password | Approved | Details              |                        |                         |
|---------|----------|----------|----------------------|------------------------|-------------------------|
| Admin   | Defcon5  | True     | <a href="#">Edit</a> | <a href="#">Delete</a> | <a href="#">Details</a> |
| Manager | manager  | True     | <a href="#">Edit</a> | <a href="#">Delete</a> | <a href="#">Details</a> |
| Guest   | guest    | True     | <a href="#">Edit</a> | <a href="#">Delete</a> | <a href="#">Details</a> |
| nobody  | nobody   | True     | <a href="#">Edit</a> | <a href="#">Delete</a> | <a href="#">Details</a> |
| turner  | calnail  | True     | <a href="#">Edit</a> | <a href="#">Delete</a> | <a href="#">Details</a> |
| turner  | turner   | True     | <a href="#">Edit</a> | <a href="#">Delete</a> | <a href="#">Details</a> |

Add User

IMPORTANT: Do NOT leave this window open since users can see passwords in plain text!!!

Editing a user:

| Edit User Form                                                             |                                                          |
|----------------------------------------------------------------------------|----------------------------------------------------------|
| User Name                                                                  | <input type="text" value="Admin"/>                       |
| Password                                                                   | <input type="text" value="Defcon5"/>                     |
| User Privledges                                                            | Administrator <input type="button" value="v"/>           |
| First Name                                                                 | <input type="text" value="Road"/>                        |
| Last Name                                                                  | <input type="text" value="Runner"/>                      |
| Email                                                                      | <input type="text"/>                                     |
| Office                                                                     | <input type="text" value="Professional Coyote Removal"/> |
| Branch                                                                     | <input type="text"/>                                     |
| City                                                                       | <input type="text"/>                                     |
| Zip                                                                        | <input type="text" value="11111"/>                       |
| Phone                                                                      | <input type="text" value="1112223333"/>                  |
| Approved                                                                   | <input checked="" type="checkbox"/>                      |
| <input type="button" value="Submit"/> <input type="button" value="Reset"/> |                                                          |

Permissions for (admin,manager,viewer)

This menu is very straightforward just edit any information you wish. The approved check box is what allows a user to use his/her username and password.

There are 3 levels of users. Admins can do anything to the site **potentially harmful things**. Managers can search and add projects only. Viewers can only use the search features.

There is no guest searching only validated accounts will be allowed to login.