Software Design Specification

February 22, 2011

Web Air & Water Dispersion Model (WebAWDM)

Version 2.2

# Egg Men Industries

*Cyle Lainson*

*Chad Ellsworth*

*Ryan Simmons*

# Table of Contents

|  |  |
| --- | --- |
| I Introduction | 1 |
| II Architecture Overview | 3 |
| III Detailed Module Description | 4 |
| 1 SCREEN3 | 4 |
| 2 Web Application | 5 |
| 3 Database | 9 |
| IV Implementation Plan | 10 |
| 1 Schedule | 10 |
| 2 Possible Changes | 11 |
| VII References | 11 |

**I. Introduction**

SCREEN3 is an air quality model that was released by the United States Environmental Protection Agency (EPA) for the purposes of predicting how the concentration of pollution given off by an emission stack will disperse across different geographic types.[1]

SCREEN3 is primarily used by companies as a preliminary report that models how pollution generated from a proposed building will affect the surrounding area. The report is then used to obtain building permits from local authorities. If the report meets or exceeds legal limits provided by EPA requirements, a more thorough modeling system is used to verify the results from the SCREEN3 model. A secondary use for SCREEN3 is during time-critical disaster scenarios. The portability of SCREEN3 permits the quick generation of models affording emergency responders the ability to make impromptu decisions.

The SCREEN3 program is available for download from the EPA website; it includes documentation and source code.[1] However, the executable included is compiled to work only on 32-bit Windows compatible systems. This is an issue since most computers are now 64-bit systems. Our group experienced this issue, as only a single member with a 32-bit system was able to run the program.

All current offerings of SCREEN3 software, including programs from Lakes Environmental and Breeze Software, are based on locally installed software. This factor limits the compatibility and accessibility of current SCREEN3 offerings to the end user. Cost is also an issue, as SCREEN3 software can be priced as high as $500.

Screen View, the Lakes Environmental SCREEN3 modeling system, has already shown that a framework can be built on top of the EPA’s SCREEN3 source code.[2] Screen View provides an interface that allows users to input data and generate graphic output. However, it lacks the portability, universal compatibility, and capability to archive data that our software will include.

Our goal is to create a web-based SCREEN3 program, meaning any operating system which supports internet browsing will be fully compatible. Initially, we thought to rewrite the SCREEN3 source code, provided by the EPA, from FORTRAN to a browser-friendly language such as Java. Time and resources have made this option infeasible. Instead, we decided to use the SCREEN3 source code in a server-side processing scheme. This presents three advantages: 1. processing will be independent from the user’s system, 2. an interface to process data from external sources has already been implemented into the code, 3. the source code is reliable, as attested to by its widespread government use. An indirect advantage to using the SCREEN3 source code is that it allows for more time to focus on other aspects of our system. Such features as an archiving system, administration tools, options for displaying results, and an intuitive user interface will greatly benefit from increased implementation time.

Our vision for this project is that a user subscribing to our program, Web Air & Water Dispersion Modeling (WebAWDM), will have the ability to use any computer or smart phone with internet access to model pollution dispersion from anywhere. Users will not be required to install any software, as all functionality will take place online. WebAWDM will be a container for multiple dispersion models, SCREEN3 being the first. By providing a system that is entirely web based, WebAWDM will provide portability and a level of functionality not available on the market.

WebAWDM will offer users with all the tools necessary to utilize the SCREEN3 model in as simple a process as possible. Input fields will be organized into segments; progression through these segments will depend on the user’s input. The data input pages will contain input forms that will transition dynamically between segments, revealing no more than a single segment at one time. Making only one segment viewable will reduce mistakes and allow users to be less overwhelmed by the length of the forms.

WebAWDM gives users two options for saving generated output. First, users will be able to download and save the output onto their local machine. Also, they will have the option of archiving their data onto the server, making it accessible from anywhere with internet access, regardless of what computer they are using. When accessing archived information, the data is automatically placed in the relevant input fields. This feature will allow users to make small adjustments to data and re-graph results simply.

The system will need to be modular and well documented. SCREEN3, being the simplest and most widely used model, will be the first of many EPA models implemented. Our goal is to make the addition of these models as easy as possible for future developers.

There are a number of challenges we will face when creating WebAWDM. First is the requirement that WebAWDM should be compatible with a large variety of platforms, and in particular iOS, the operating system for the iPhone and iPod Touch. In order to work out any issues which may arise, we have given ourselves additional time at the end of each major phase of development.

Another concern is that we are not environmental engineers, and as such we have, at best, a tenuous understanding as to the inner workings of SCREEN3. We will be building the existing SCREEN3 program from the EPA into our own program to help alleviate this problem. Lacking comprehension of the variables composing SCREEN3 will also create a testing dilemma, as we will not be able to tell the difference between valid from invalid inputs, or know if a particular output is correct based on the corresponding input. As such, testing results will be done against example data sets from Lakes Environmental’s Screen View program.

Our last difficulty is that we are creating a primarily web-based application. Each of the team members has experience limited to a semester in web programming at NAU. WebAWDM will be more complicated and intricate then anything we have ever attempted. In the event that we need assistance, we will first consult online tutorials, and if further issues arise, web programming professor Darryl Brown. We are confident that, by taking such steps, WebAWDM will be a complete, successful system.

**II. Architecture Overview**

Figure 1: General Architecture

Figure 2: Connections

This system will consist of three main entities, as shown in Figure 1: the web application, the SCREEN3 modeling program, and the database. Figure 2 details the interactions between these entities.

The web application will act as the system’s user interface. This is where users will log into the system, employ the SCREEN3 model, access previously archived data, increase their subscription, and change their password. Along with general users, there will be an additional administrator class of users which hold all the abilities of general users, along with the ability to access all user data with a limited ability to change it. Administrators will have the power to extend subscriptions, reset a user’s password, add or delete users, and also delete any archived data.

The SCREEN3 model will take in a .DAT file, which will be generated from input data, and produce a text file containing ASCII output data. This data will be sent back to the web application, where the user will be able to view it in a new window. Users can request to view a graphical representation of the data, which will also appear in a new window.

The database will hold all necessary data that needs to be saved for later use. This includes user names, passwords, archived input data, and subscription information. Administrators will be able to manage the database through the web application, however only webmasters will have direct database access.

# III. Detailed Module Description

**1. SCREEN3**

This section will focus on the design of the SCREEN3 integration system. The system will be event based, and contain two separate modules. These modules, as well as the SCREEN3 model, will be located and run on the server, taking in data from the web application and sending output back.

1.1. DAT Generator

The DAT generator will wait until a text document is received and translate the data within the file into a .dat file, which will be sent into the SCREEN3 program. The web application’s Models page will then pull the resultant ASCII file for user access.

1.2. Graph Maker

The graph maker will be called when the user requests to view a graph from the Models page. This will take the ASCII file from the page, create a graph based on that data, and send the graph back to the web application’s Graph page.

**2. Web Application**

This section describes the WebAWDM web site, which acts as a user interface for the system.

2.1. Main Menu

Logo

Username

Password

Forget Password?

Not a member?

Log In

Home

Contacts

About

Figure 3: Not Logged In

The menu will be located at the top of each page, containing links for users to navigate throughout the site. The links which appear in the menu will vary depending on the user’s login status and access level. If the user is not logged in, the title bar will display Figure 3, where he or she will have the option to view the Home, About, and Contact pages; as well as log in, access the Forgot Password page, or access the Subscription page from the menu at the right.

Welcome <user>

Extend Subscription?

Logo

Archive

Models

Contacts

About

Log Out

Admin Page

Home

Figure 4: Logged In

Figure 3: Models Page

Figure 3: Models Page

If the user is logged in, Figure 4 will be displayed, which extends access to the Models and Archive pages. The user will be given the ability to further extend his or her subscription, or log out through the menu at the top right.

If they are an administrator they will be given an additional link which will give them access to the Admin page.

For each page the menu giving access to the other main pages will give the appearance of folder tabs, where the active page (in this example Home) will have its corresponding tab selected. The logo for the project will be in the top left, slightly behind the WebAWDM title graphic.

2.2. Models

TITLE

Model

Model page

Data entry name

Save Online

Save Locally

View Data

View Graph

Next

Finish

Previous

Figure 5: Models Page

The Models page, as seen in Figure 5, will consist of a drop down menu where users will select the model they wish to access, after which the corresponding model page will display beneath it. Underneath the model page there will be three buttons labeled “Previous”, “Next”, and “Finish”. The “Previous” button will move the model page down, “Next” will move the model page up, and “Finish” will send a request to enter the model information to the program. This will prevent an overwhelming amount of information from appearing on the page at once, and will hide some of the dynamic aspects of the model page. If the information is valid, it will be sent into the model, and upon the modeling program’s completion the generated output will appear in ASCII format in a new window. Also, two new buttons will appear on the page, labeled “View Graph” and “View ASCII”. The “View ASCII” button will simply re-open the window with the ASCII data if the user has minimized or closed it. The “View Graph” button will send a copy of the output back to the modeling program, which will create a graph based on the data and send it to the web application, where it will also appear in a new window. Underneath these two buttons will be a data entry text box and a button which will allow users to create a title for their input data and then save it to the server.

2.3. SCREEN3 Input

The SCREEN3 input page is a form containing fields requesting the information necessary to run the model. It will be a dynamic webpage situated within the Models page. The input fields which appear will change based on the information the user has already filled. For example, choosing a point source requires users to give the stack’s gas exit temperature, while a flare source does not require this information.

2.4. Archive

TITLE

Name

Source

Model

Date

12/34/5678

Point

SCREEN3

Data 1

Flare

Some Other

Data 2

09/87/6543

Data 3

SCREEN3

Volume

34/56/7890

Delete All Checked

Check All

Figure 6: Archive Page

The Archive page, as shown in Figure 6, will consist of a list of every submission the user has saved to the server. There will be 4 options available by which to sort the data: Name, Model, Source, and Date. Name will sort the submissions by the title the user gave it when it was saved. Model will allow the user to select what type of model he wishes to browse and only show submissions using that model. Source will only show users saved data which used a particular source type. Date will sort the submissions by the time that they were made.

Each submission will display the Name, Model, Source and Date along with a link to the Models page on the name. Accessing the Models page in this manner will automatically enter all the data they entered for that submission into the corresponding fields. They will then be easily able to re-graph the model, access the ASCII text file, or make changes to the entered fields.

There will be a check box next to each submission and a button at the bottom allowing the user to delete all checked submissions.

2.5. Admin

TITLE

Create New User

0

User3@a.com

34

User2@a.com

3477

User1@a.com

Username

# of Entries

Email

Subscription

User3

User2

User1

09/87/6543

12/34/5678

34/56/7890

Delete All Checked

Figure 7: Admin Page

The Admin page will consist of a list of all the users with their username, e-mail address, the date their subscription ends, and how many submissions the user has made. Each username will be a link to their User page. There will be a link at the top to the Create User page.

2.6. Create User

The Create User page will be a simple data entry page with four fields: username, password, email and subscription. Underneath that there will be a button to create the user, which will add the user to the Database.

2.7. User

The user page will show the username, email, and subscription information of the user. The subscription can be extended by using a data entry form at the right. The user’s archived data submissions will appear below the user information in the same format as the archive page.

2.8. Graph

The Graph page will only show up after the user has used a model to create a graph and then presses the View Graph button on the Models page. This will display the graph in a new window for the user to view and/or save onto local memory.

**3. Database**

The database is a MySQL server which will contain information for the system. It will hold usernames, emails, passwords, subscription information, administrator status and archive data. The primary purpose is to allow the web application to easily pull the relevant information. The database design is diagrammed in Figure 8.

Figure 8: Entity-Relationship Diagram

# The database will consist of two entities: User and Data. Each user’s entry will consist of a unique username which will act as the primary key, a hash of the user’s password, an e-mail address, the user’s status (administrator or general user) and the end date of the user’s subscription.

# Each user will be able to archive multiple data entries. Data is a weak entity, as its primary key will be a combination of the name of the data and the username of the user which archived it.

Also stored in each Data database entry is the model and source used to generate the data, the date the data was archived, and of course the .dat file.

# IV. Implementation Plan

**1. Schedule**

Our first goal is to create a fully functional website by the beginning of March. This includes all the forms and databases required to house the SCREEN3 program, along with security concerns such as the user and administrator systems and login functionality.

After the site is finished, the next step is to tackle SCREEN3 functionality. The data modeling program will first be developed separately from the website, and then incorporated in once it is running smoothly. Since the SCREEN3 model already exists in a scientifically correct form, much of the work to be done with the program will focus on ensuring the model’s FORTRAN code will be integrated into the Java program successfully. Generating accurate graphs based on SCREEN3’s ASCII output will be another top priority. We hope to have the modeling program fully functional by March 15th, and then incorporated into the website by March 28th. Then, we will use the rest of our available time to thoroughly test our system and work on documentation.

Figure 9: Gantt Chart

**2. Possible Changes**

If there is any specific milestone which will probably take longer than the time allotted here, it would probably be either the form functionality or the modeling program functionality. If we do need more time to work on either of these sections, we have left ourselves an additional week before the point in time when we would like to have the entire project ready for testing, April 5th. If this is still not enough time, we can certainly lessen the amount of time we have set aside to incorporate the modeling program into the website. It probably will not require two weeks of work to get a fully functional Java program to work with a fully functional website. That being said, we still hope to meet each milestone promptly and have plenty of time to ensure the program and website is working well together.

# V. References

United States. Environmental Protection Agency. Screening Models.17 Aug. 2010. Accessed

17 Feb. 2011 <http://www.epa.gov/scram001/dispersion\_screening.htm#screen3>.

“Screen View.” Lakes Environmental. 2011. Accessed 17 Feb. 2011

<http://www.weblakes.com/products/screen/index.html>.