

Project Plan

for the

Construction Department of Southern Illinois University Edwardsville

by

Bryan Allen – Quality Assurance

Daniel Grote – Customer Representative

Zach Smith – Scrum Master

of

CS425 / CS499 Senior Project

Concrete Drying Application Team

CS-PLAN

Revision 1.0.2

As Of: 20 January 2015

Change Log:

<u>Revision</u>	<u>Change Note(s)</u>
1.0.1	Made modifications for sprint 1 preparation phase, and made changes as suggested by Dr. Mayer
1.0.2	Removed usability study and updated sprint schedule

Table of Contents

1	Estimation	1
2	Resources	2
3	Scheduling.....	2
4	Communication	3
5	Quality Assurance	4
6	Exit Strategy	5
6.1	425 Exit Strategy	5
6.2	499 Exit Strategy	6

Figures

Figure 1: High Level Architecture	1
Figure 2: Schedule	2

Tables

Table 1: Resources	2
Table 2: Risk Table	3
Table 3: Test Plan	4

1 Estimation

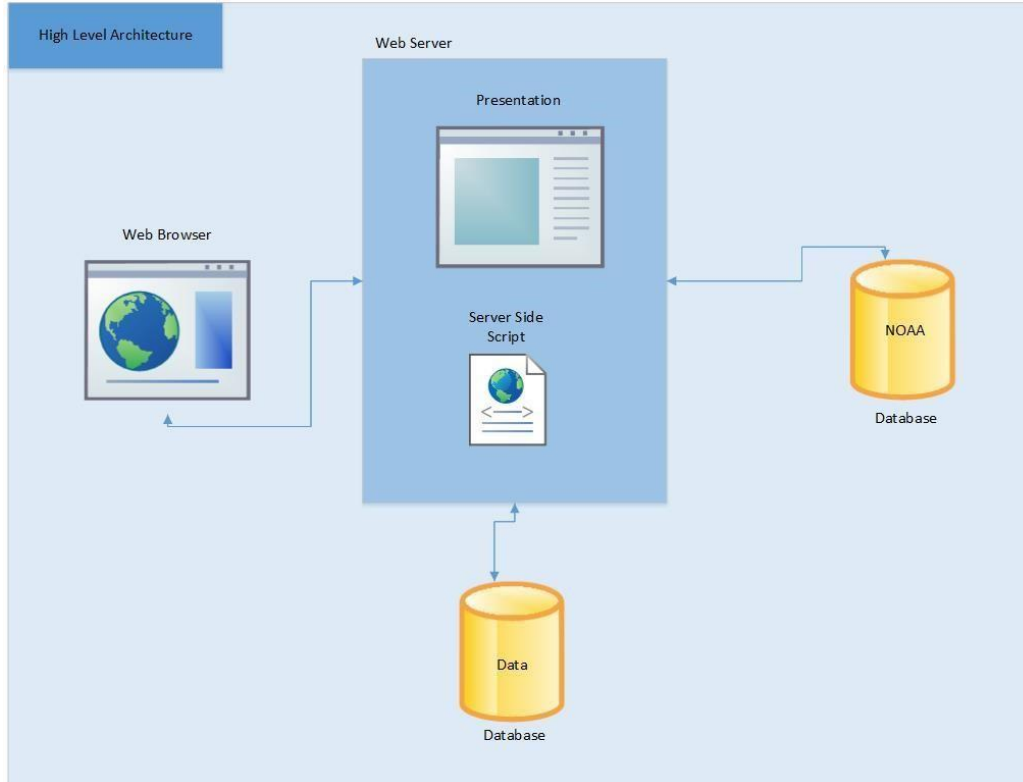


Figure 1: High Level Architecture

To estimate the time need for the overall project it has been divided into four parts. The first part is to complete the displaying of the evaporation rate and graph. The next is to create login functionality. The third thing is to add the notification system. Following that, ways of viewing statistical information about the site would be created.

Milestones:

Display Evaporation Rate

Account

Notifications

Statistical Data

Total = 107 person hours

Sprints:

- | | |
|---|--------|
| 1) Get data from NOAA | Oct 27 |
| a) Research Web Services and API's | |
| b) Pull weather data from NOAA | |
| c) Create function to calculate evaporation rate | |
| d) Test Cases | |
| 2) Create core UI | Nov 10 |
| a) Make a start page (I was thinking something simple like the main google page, except you just type in a zipcode) | |
| b) Show results from calculations in a graph | |
| c) Make a graph display page | |
| 3) Create user account UI & Code the user accounts db (userID, email pw) | Jan 20 |
| a) Edit account page | |
| b) Projects page | |
| c) Notifications page | |
| d) Learn secure procedures for handling passwords | |
| 4) Code the user project db (userID, projectName, zipcode) | Feb 2 |
| 5) Code the user notifications db (userID, zipcode, time, lastEvapRate) | Feb 23 |
| a) When a notification is added, add zipcode to the global projects db if not in there | |
| b) Update data from global projects db | |
| c) Send notification on change of lastEvapRate | |
| 6) Code the global projects db (zipcode, time, other weather data) | Mar 16 |
| a) Update every zipcode periodically (hour, 3 hours?) | |
| 7) Create admin UI | Mar30 |
| a) Display usage statistics | |
| 8) Empty Sprint for any added requirements | Apr 13 |

4 Communication

In addition to using Bitbucket, we will be using Dropbox for some file management. Also, we use Google Hangouts for group communication, and TeamViewer to look over documents and edit them while we are not altogether. To communicate with our customer we use email, and provide updates after each sprint. We also show working prototypes when available.

5 Quality Assurance

Table 2: Risk Table

No.	Risk	Likelihood	Impact	Score	Mitigation Strategy	Contingency Plan
1	no prior security knowledge	10	10	100	N/A	Learn security
2	limited web API knowledge	10	9	90	N/A	Learn web API
3	fail to meet deadlines	4	8	32	try to space our milestones out in a way that gives us enough time complete them, meet at least once a week to make sure everyone is on track	re-evaluate our milestones and adjust them
4	customer adds requirements	5	4	20	get all the requirements beforehand and add it to the process spec	change the process spec and project plan to account for new requirements
5	cannot get siue email for notification service	4	4	16	trying now to get one from ITS	use a free account from another email service
6	NOAA permanently goes offline	1	10	10	write code with low coupling so that it is easily able to use another weather API	Find a different weather service
7	Project data gets lost	1	10	10	store on our local computers in addition to using bitbucket	reupload latest version we have to minimize lost work

The higher the number the more chance and effect the risk has.

Table 3: Test Plan

What is being Tested	Type of Test	Description
Customer Satisfaction	Acceptance testing	We will have the customer use the product when he is going to be working on a project to see if it conforms to his requirements. project to see if it conforms to his requirements.
Navigation	System testing	Ensure overall easy navigation and easy to understand directions
Calculating probability of shrinkage cracks	A/B testing	We will run numerous calculations manually and by the program to ensure that results are accurate.
Data retrieval	Performance testing	We will run data retrieval at varying times of the day with different network conditions in order to record how long it takes to pull data from NOAA.

6 Exit Strategy

6.1 425 Exit Strategy

Our teams exit strategy is to have a core component finished. That core component will have the functionality to be able to enter a location then having a web service gather weather data from NOAA for every hour for the next seven days. Then for each hour perform the calculation to determine the evaporation rate which determines the probability that concrete shrinkage cracks will form. This data will then be presented in a line graph. The line graph will have 3 colors red, yellow, and red where red has the highest risk, yellow has a moderate chance, and green means minimal/no chance of shrinkage cracks to form. The line graph will have different views such as week view and day view and specific weather data that was used for a calculation will be able to be seen by clicking on the line graph for a specific hour of the day.

6.2 499 Exit Strategy

The second semester exit strategy includes tests and demonstrations. When implementing the rest of the requirements (Create a database, create/edit/delete an account, forgot password, add/edit/delete notification, add/modify/delete project, and web site usage data) we will perform performance and system testing on notifications to determine if weather data is being updated correctly when needed and if the notifications is correctly notifying the user. Also, we will perform compatibility testing to make sure that for the 3 major browser the software works correctly and looks similar on each browser.

After the rest of the requirements are done being implemented we will perform installation testing. One of the last things we do is perform acceptance testing which will determine if the client accepts the software and all of the clients requirements are met and satisfied. Guides will then be made on how to view the web site usage statistics by using the admin account.