

Project Specification

for the

Construction Department of Southern Illinois University Edwardsville

by

Bryan Allen

Daniel Grote

Zach Smith

of

CS425 / CS499 Senior Project

Concrete Drying Application Team

CS-SPEC

Revision 1.0.1

As Of: 17 September 2014

Change Log:

<u>Revision</u>	<u>Change Note(s)</u>
1.0.1	<ul style="list-style-type: none">• Added verification section• Made grammatical corrections

Reviewed and Approved By:

NameSignatureDate

Table of Contents

1	INTRODUCTION	1
1.1	Business and Domain Description	1
1.2	Concept of Proposed System / Scope	1
1.2.1	Operational Concept and Scenario	1
1.3	Product Overview	1
1.3.1	Product Perspective	1
1.3.2	Product Functions	2
1.3.3	User Characteristics	2
1.3.4	Limitations	2
1.4	Definitions	2
1.5	Stakeholders	4
2	REFERENCES	4
3	SPECIFIC REQUIREMENTS	4
3.1	External Interfaces	4
3.2	Functions	5
3.2.1	Agile Use Cases	5
3.3	Usability Requirements	7
3.4	Performance Requirements	8
3.5	Logical Database Requirements	8
3.6	Design Constraints	8
3.7	Software System Attributes	9
3.8	Supporting Information	9
3.8.1	Sample input/output formats	9
3.8.2	Description of the problem to be solved by the software	10
4	VERIFICATION	10
4.1	External Interfaces	10
4.2	Functions	11
4.2.1	Agile Use Cases	11
4.3	Usability Requirements	11
4.4	Performance Requirements	11
4.5	Logical Database Requirements	11
4.6	Design Constraints	11
4.7	Software System Attributes	12
4.8	Supporting Information	12
5	APPENDICES	12
5.1	Assumptions and Dependencies	12

5.2 Acronyms and Abbreviations 12**Figures**

Figure 1: Use Case Diagram 5

Figure 2: Evaporation Rate Example Output 10

Tables

Table 1: Stakeholders..... 4

1 INTRODUCTION

1.1 Business and Domain Description

This purpose of this document is to define the requirements of the Concrete Drying Application, and provide analysis. The business of pouring concrete relies on the ability to determine whether or not shrinkage cracks will occur by calculating the evaporation rate. To calculate the evaporation rate, a nomograph is used in conjunction with weather data provided by the NOAA.

1.2 Concept of Proposed System / Scope

We will forecast the probability of shrinkage cracks in an easier and more efficient way. We will automatically pull weather data from the NOAA databases and calculate the evaporation rate for every hour of every day for up to seven days. Then, we will display this information in the form of a graph that the user will be able to zoom in and out of as well as textual data.

1.2.1 Operational Concept and Scenario

- The application will be launched like any other iOS application, which is by tapping the appropriate icon.
- Once the application is running you will be on the home screen of the application.
- On the home screen of the application the user will be able to create, edit or delete projects.
- When a project is created the user will be able to enter project specific data (such as location and name of the project).
- After the project is created the user will be able to view the evaporation rate for up to seven days.
- The user will also be able to select a specific time for which the application will notify them if the evaporation rate changes.

1.3 Product Overview

1.3.1 Product Perspective

This project is going to rely on the NOAA site for data, along with some built in features of iOS devices (such as GPS and Notifications). When designing the user interface we plan on following the iOS UI guidelines as specified by Apple.

- a) system interfaces: access stored database of saved weather data
- b) hardware interfaces: not applicable. (The app does not control any hardware components)
- c) software interfaces: NOAA database
- d) communication interfaces: Wi-Fi, cellular network, GPS
- e) memory: should use little memory, managed by iOS
- f) operations: pull data from NOAA, calculate evaporation rate, display calculated information
- g) site adaptation requirements: in order to get fully updated information, the user must be connected to the internet either by Wi-Fi, or cellular network

1.3.2 Product Functions

There will be three major functions in application. The first will be pulling the required data off the NOAA site. The next is calculating the probability of shrinkage cracks. Lastly the application will create an easy to read graph for the users.

1.3.3 User Characteristics

The targeted users will be construction managers that have access to an iOS device. The user will be familiar with how to use an iOS device, and have experience reading graphs. The user's location will vary, but will not matter as long as internet access is available.

1.3.4 Limitations

- This application will not be able to update data used for calculations if internet access is not available.
- The application can only retrieve updated weather data when the NOAA databases have been updated.

1.4 Definitions

Concrete Drying - waiting for the concrete to dry

Concrete Floor Pours - pouring concrete to make a floor

Concrete Project - a specific concrete job site (multiple projects can exist)

Concrete Shrinkage Cracks - cosmetic and/or structural cracks that appear in concrete due to evaporation rate

Contractors - a person or company that undertakes a concrete floor pour

Construction - process of preparing and forming something

Evaporation Rate - rate at which water evaporates / used to forecast concrete shrinkage crack probability in concrete floor pours

Forecast - predicting the evaporation rate

Humidity - amount of water vapor in the atmosphere (variable used to calculate evaporation rate)

iOS Devices - iPads and iPhones

Location - the site where the concrete is going to be poured

Materials - the substances that make up the concrete (variable used to calculate evaporation rate)

Mobile Application - an application on a mobile device (ex: iPhone)

National Oceanic and Atmospheric Administration - the name of the organization chosen to gather weather information from

Nomograph - chart used to manually predict the evaporation rate

Notification - an alert that brings a change to the user's attention

Percent Cloud Cover - weather term to measure the amount of sunshine visible (variable used to calculate evaporation rate)

Projects - a specific job site (multiple projects can exist)

Temperature - measurement of how hot or cold and area is (variable used to calculate evaporation rate)

User Interface - what the user sees and interacts with

Wind Direction - the direction the wind is blowing (variable used to calculate evaporation rate)

Wind Speed - the speed at which the wind is blowing (variable used to calculate evaporation rate)

1.5 Stakeholders

Table 1: Stakeholders

People	Organization	Relationship
Bryan Allen	SIUE Computer Science Student	Developer
Daniel Grote	SIUE Computer Science Student	Developer
Zachary D. Smith	SIUE Computer Science Student	Developer
Mark Grinter	SIUE Construction Dept.	Client
Dr. Mayer	SIUE Computer Science Dept.	Oversight

2 REFERENCES

Mayer, Gary “Semi-Agile Software Engineering (SAGE) Process Specification”
Revision 2.1.3 18 August 2014: 1-40. Print

Mayer, Gary “Semi-Agile Software Engineering (SAGE) Deliverable Standard”
Revision 1.0 06 September 2013: 1-12. Print

ISO/IEC/ IEEE “Systems and software engineering — Life Cycle Processes —
Requirements Engineering” First Edition 1 December 2011: 1-83. Print

Uno, Paul “Plastic Shrinkage Cracking and Evaporation Formulas” ACI Materials
Journal, 1998: 365-75. Print

3 SPECIFIC REQUIREMENTS

3.1 External Interfaces

Not Applicable

3.2 Functions

3.2.1 Agile Use Cases

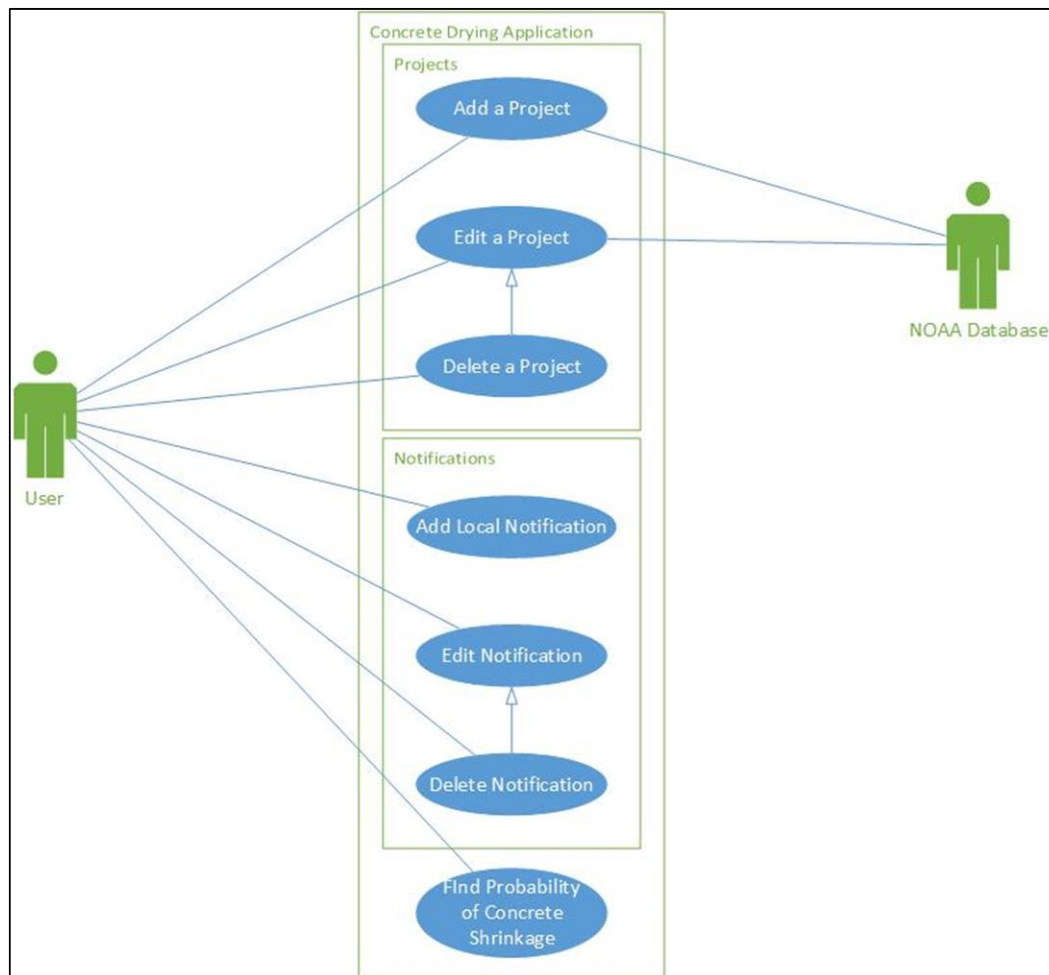


Figure 1: Use Case Diagram

Add a Project

Trigger: User wants to create a new project to forecast concrete shrinkage

1. User selects add new project button
2. User enters a name for the project
3. Location defaults to current location
 - a. Or user changes the location to the desired location
4. User enters project specific data 5
5. User submits form

Edit a Project

Trigger: User wants to change the project name or project specific data

1. User selects the edit project Button
2. User selects the edit information button on the project that needs changed
3. User changes project specific data
4. User submits changes

Delete a Project

Trigger: User wants to delete a project

1. User selects the edit project button
2. User selects the delete project button on the project that needs deleted

Add Local Notification

Trigger: User wants to be notified if the probability of concrete shrinkage dramatically changes on a user specified date

1. User adds a new project or chooses an existing project
2. Color coded line graph is displayed over a period of time for the user to see the probability of concrete shrinkage
3. User selects add notification
4. User selects a date
5. User chooses the parameters of when to be notified (Example: If the probability falls under X%)
6. User submits form
7. Notification(s) is displayed under the line graph

Edit a Local Notification

Trigger: User wants to change a local notification

User selects the project with the notification that needs changed

1. Color coded line graph is displayed over a period of time for the user to see the probability of concrete shrinkage
2. User selects the edit notifications button

3. User selects the edit button on the notification that needs to be changed
4. User changes the notifications date or parameters.
5. User submits changes

Delete a Local Notification

Trigger: User wants to delete a local notification

1. User selects the project with the notification that needs to be deleted
2. Color coded line graph is displayed over a period of time for the user to see the probability of concrete shrinkage
3. User selects the edit notification button
4. User selects the delete button on the notification that needs to be deleted

Find probability of concrete shrinkage

Trigger: User wants to find the probability of concrete shrinkage

1. User adds a new project
 - a. User selects add new project button
 - b. User enters a name for the project
 - c. User enters project specific data
 - d. User submits form
 - e. Color coded line graph is displayed over a period of time for the user to see the probability of concrete shrinkage
2. User looks at an existing project
 - a. User selects one of the existing projects
 - b. Color coded line graph is displayed over a period of time for the user to see the probability of concrete shrinkage

3.3 Usability Requirements

The application should be quick and easy to learn.

- 90% of users should learn how to use the application is five minutes or less.

The application should be easy to navigate.

- 90% of users should be able to navigate to a specified page in less than one minute.

The application output should be clear and understandable.

- 90% of targeted users should understand what the output means and how to use this information effectively.

The applications output should be trusted by the user.

- The applications formula calculations should be right 100% of the time.

3.4 Performance Requirements

- 95% of data retrieval shall take place in less than five seconds
- Formula calculation should take less than 1 second to complete

3.5 Logical Database Requirements

- Project
 - Name
 - Location (zip code)
 - For each hour of each day out to 7 days:
 - Surface Wind
 - Relative humidity
 - Atmospheric Temperature
 - Material Temperature
 - Percent Cloud Coverage

3.6 Design Constraints

- User must have Internet access to find up to date weather information
- When using notifications user must have internet access so the user can get notified correct

3.7 Software System Attributes

- Reliability – The software must work all the time when there is internet connection and the NOAA database tables are live. If there is no internet connection or the NOAA database tables are currently unavailable the software will notify the user.
- Availability – Software should be readily available when the user has access to the Internet.
- Security
 - Check data integrity for critical variables
 - Validate data entry
- Maintainability – Application's modules should be loosely coupled so that new requirements can be easily added.
- Portability – Application must run on all iOS devices. The interface must be similar and easy to use on the varying screen sizes.

3.8 Supporting Information

3.8.1 *Sample input/output formats*

Input would be textboxes where user enters into textboxes.

The output would be a color-coded line graph. Green would mean it is safe to pour. Yellow would be caution (Take caution when pouring). Yellow would also give tips/steps on how to reduce the chance of concrete shrinkage. Red would mean do not pour.

These items should be considered part of the requirements.



Figure 2: Evaporation Rate Example Output

3.8.2 Description of the problem to be solved by the software

The software would automate the process of calculating the evaporation rate for a given location and time, then display the information in an easy to read format.

4 VERIFICATION

The client verifies this document to ensure that the proposed requirements are the desired requirement.

4.1 External Interfaces

Not Applicable

4.2 Functions

4.2.1 Agile Use Cases

Each use case must be represented and each use case needs to contain the necessary functionality of the application. The use cases need to be verified to see whether or not the needed functionality is present.

4.3 Usability Requirements

Bring in ten test subjects and introduce them to the application interface.

- Leave them to navigate the interface for five minutes. After that ask them to do all the basic functions of the application and find the number of them that can accomplish these tasks.
- Ask them to go a specific page of the application in under a minute and find the number of them that can accomplish this.
- Ask them if they can read and understand the output of the application and find the number of them that can read and understand.

Do the calculations manually and compare the results to the results of the applications formula.

4.4 Performance Requirements

- Run the data retrieval 100 times and find the number of times the data is retrieved in less than five seconds.
- Run the formula calculation 100 times and find the number of times the calculation took less than one second to complete.

4.5 Logical Database Requirements

Perform testing to determine if the proper projects and local notification are in the database and are correctly displayed.

4.6 Design Constraints

Perform testing to verify the application notifies the user when there is no internet access and/or there is not up to date data being used in the application.

4.7 Software System Attributes

- Reliability – See section 4.4
- Availability – Test the application by running it for a set period of time and recording the number of crashes (Availability testing).
- Security
 - Test to see if weather variables have changed since last use.
 - Test variable validation
- Maintainability – Perform preventive maintenance to reduce future maintenance cost.
- Portability – Perform a usability study to see if the application is similar and easy to use on the different iOS devices.

4.8 Supporting Information

Perform a usability study to determine if the input and output of the software is intuitive and easy to use. (See section 4.3)

5 APPENDICES

5.1 Assumptions and Dependencies

- Users will have internet access
- Users have iOS device(s)
- NOAA database is live and has the forecast predictions
- Application is permitted to use device GPS
- Application is permitted to show local notifications

5.2 Acronyms and Abbreviations

GPS – Global Positioning Service

iOS – iPhone Operating System

NOAA - National Oceanic and Atmospheric Administration

UI – User Interface

Wi-Fi – Wireless local network