Software Requirements Specification Template

BRICK FINDER

Software Requirements Specification

April 02, 2019

Team Members

Bhavika Padidala

Priyanka Bonam

Wendy Eloe

Subhodh Bhargav Lakhinana

Durga Prasd Kallem

Sachin Setty

Submitted in partial fulfilment

Of the requirements of

CSIS 44691-01 Graduate Directed Project 1

**Table of Contents**

**Table of Contents Page Number**

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations
   4. References
   5. Overview
2. General Description
   1. Product Perspective
   2. Product Functions
   3. User Characteristics
   4. General Constraints
   5. Assumptions and Dependencies
3. Specific Requirements
   1. External Interface Requirements
      1. User Interfaces
      2. Hardware Interfaces
      3. Software Interfaces
      4. Communications Interface
   2. Functional Requirements
   3. Non-Functional Requirements

3.3.1. Performance

3.3.2. Reliability

* + 1. Availability
    2. Security
    3. Portability
  1. Inverse Requirements
  2. Design Constraints
  3. Logical Database Requirements
  4. Other Requirements
  5. Prototypes (for complete project)

1. Design

4.1. GUI

5. Technical Manual

5.1 Search Page

5.2 Map Page

5.3 print Functionality

5.4 Database Functionality

5.5 Security Issues

6. End User Manual

6.1 Search page

6.2 Map Page

6.3 Print Functionality

**UNIT 1: INTRODUCTION**

* 1. **Purpose**

The purpose of the project is to help a donor search for his/her brick location in the Hughes Field House.

* 1. **Scope**

The scope of the project is:

* Save the time of the donor.
* Locate the maps that points to the exact zone number.
  1. **Definitions, Acronyms, and Abbreviations**

**Visual Studio code:** Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring.

**Bootstrap:** It is a library we used to make the page responsive.

**Data Folder**: It has the Excel sheet from where we are reading the data.

**Lib Folder:** It has all the libraries that are generated automatically when we push the code in to the GitHub repo.

**Img Folder:** It has all the images that we are using in the project.

* 1. **References**

*Visual Studio code developer documentation*. (2019). Retrieved from:

<https://code.visualstudio.com/docs>

* 1. **Overview**

The Brick Finder system will help the donor search for the location by entering either the name or inscription in the text field. Once the donor clicks on the search button, they will see a couple of rows with the matching results. They can find their name and by clicking on that particular row they will see a map with the respective zone number on it and also a brick/paver with inscription on it. Also, if the donor wants to print the page they can print it by clicking on the print button which will appear on the page.

**Unit 2: GENERAL DESCRIPTION**

* 1. **Product Perspective**

The product perspective is to make donors search for their brick/paver in the Hughes Field House. This solution is very helpful for donor and the administrator. They can easily identify the location and take a print of the map.

* 1. **Product Functions**

The product functions are divided into two tasks: admin and users.

**Admin tasks:**

1. Add new donors in to the excel sheet.
2. Update or remove donor information.

**User tasks:**

1. Search for the location by entering name or inscription
2. View the results in a tabular form
3. View the location map
4. Print the location map and a brick/paver with inscription on it.
   1. **General Constraints**

* Only admins can manage the Excel sheet.
* Donors can only search for the location and print the map (if needed).

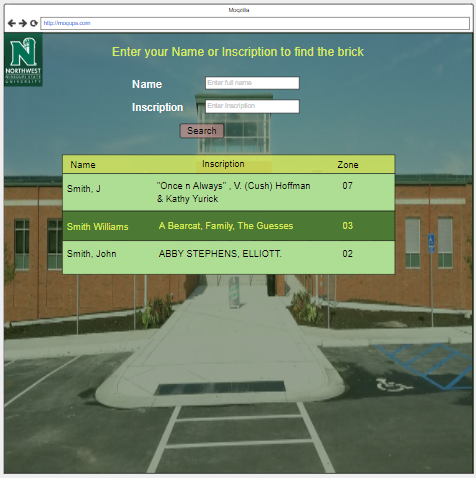
**Unit 3. SPECIFIC REQUIREMENTS**

* 1. **External Interface Requirements** 
     1. **User Interfaces**

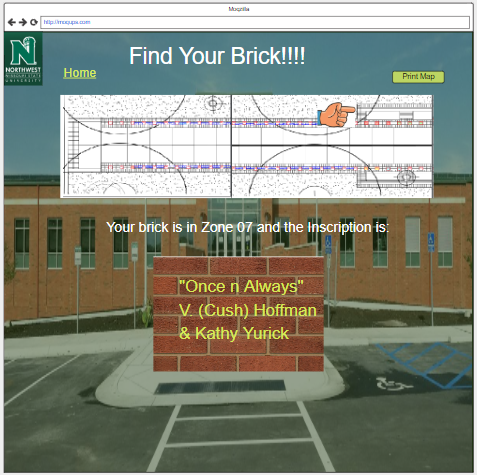
User interfaces used in this project are mentioned below, Search screen, and Map Screen. We also made the web page responsive so that it looks good even on a Mobile.

**Web Site:**

Search Screen:

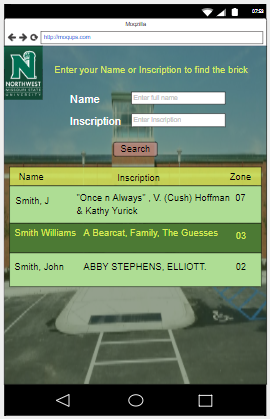
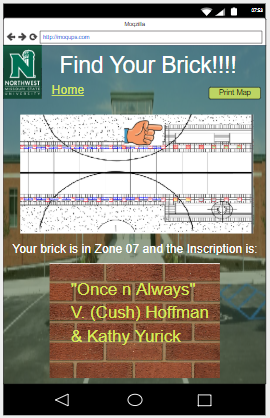


Map Screen:



**Mobile site**

Search screen: Map screen:

* + 1. **Hardware Interfaces**

Our hardware interfaces are iPhone, iPad, MacBook, Mac Pro, Android devices. Memory should be GB. Operating system using is Windows. Processor is 2.3GHz dual-core Intel Core i5 7th Generation.

* + 1. **Software Interfaces**

As for the programming environment, OS is a windows system and as such it comes with most of the tools you need to develop. The IDE must be downloaded separately from Visual Studio, it’s called "Visual Studio Code". The current version of the IDE is 1.19.2 but version 1.33 is also available out.

**3.1.4 Communications Interface**

**Google Docs, sheets and slides:**

Google Docs, sheets and slides lends itself to collaborative projects in which multiple authors work together in real time from geographically diverse locations. All participants can see who made specific document changes and when those alterations were done. Because documents are stored online and can also be stored on users' computers, there is no risk of total data loss as a result of a localized catastrophe. However, the Internet-based nature of Google Docs, sheets and slides has given rise to concerns among some authors that their work may not be private or secure.

We use this for creating project documents, sheets and slides for presentation. It is reliable for team-work.

**Share Point:**

SharePoint is a web-based collaborative platform that integrates with Microsoft Office. Launched in 2001, SharePoint is primarily sold as a document management and storage system, but the product is highly configurable, and usage varies substantially between organizations.

We use share point to work with team on same platform as Microsoft word.

**Outlook:**

It is Microsoft e-mail service. We use outlook to send and receive mails. Used to contact team members.

* 1. **Functional Requirements**

**User Modules**

1. Search

The donor should be able enter the name or inscription and should be able to search their bricks/pavers.

1. Map

The donor should be able to view the map with respective zones by clicking on any of the search results.

1. Print

Donor should be able to print the page that has the map and the image of a brick/paver with inscription on it.

**3.3 Non-Functional Requirements**

**3.3.1. Performance**

This is to check the speed of the page that means how fast the page is responding to the actions. Generally, every action should not take more than 20 seconds to respond. In this application,the user should not feel any delay in response when he performed the series of action through the page.

**3.3.2. Reliability**

Reliability is the percentage of time that the page works correctly without any failures and to deliver the results properly. For our page it is the amount of time the page can run process of searching the information based on the input given and displaying the results. It should be 99 percentage in general.

**3.3.3. Availability**

Availability means the provider systems should meet the agreed availability targets like giving the exact maps with zone appropriate zone numbers on it.

**3.3.4. Security**

Client permitted us to use a public repository to the Excel data, as there is no much detailed information about the donors, like the address, phone number etc. so, there are no security issues in making the data public.

* + 1. **Portability**

Portability checks for whether the web page works on both the browser and the mobile. Our page works on both and we used bootstrap library to make the page responsive, so the user can open our page from a browser as well as mobile.

**3.4 Inverse Requirements**

Inverse requirements state the requirements what the system will not do. It describes the constraints on allowable behavior. In most of the cases, it is easier to state that certain behavior must never occur than to state requirements guaranteeing acceptable behavior in all circumstances. It depends mostly on the client.

**3.5 Design Constraints**

We need to be able to identify the constraints involved in designing a website before diving into the few basic principles in page design that are common to most effective sites.

1. The fundamental constraint is fixing the objects that will appear on a viewer’s screen by thinking from the viewers point.
2. We should create page designs that bend but don’t break when the users viewed in different environments.

**3.6 Logical Database Requirements**

We did not use any database server in order to store the data. Our application is completely a client side application. It will be easy for the client to update the excel sheet any time easily.

**3.7 Prototypes (for complete project)**

Prototype is an early sample, model, or arrival of an item worked to test an idea or process or to go about as a thing to be repeated or learned from.

**Categories of Prototypes:**

**Proof-of-Principle Prototype** serves to verify some key functional aspects of the intended design, but usually does not have all the functionality of the final product.

**A Working Prototype** represents all or nearly all of the functionality of the final product.

**A Visual Prototype** represents the size and appearance, but not the functionality, of the intended design. A Form Study Prototype is a preliminary type of visual prototype in which the geometric features of a design are emphasized, with less concern for color, texture, or other aspects of the final appearance.

**A User Experience Prototype** represents enough of the appearance and function of the product that it can be used for user research.

**A Functional Prototype** captures both function and appearance of the intended design, though it may be created with different techniques and even different scale from final design.

**A Paper Prototype** is a printed or hand-drawn representation of the user interface of a software product. Such prototypes are commonly used for early testing of a software design, and can be part of a software walkthrough to confirm design decisions before more costly levels of design effort are expended.

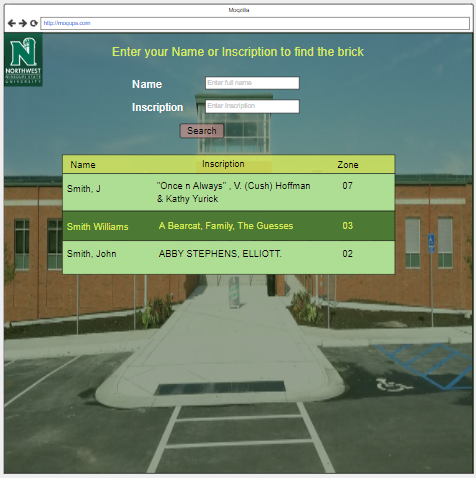
**UNIT 4. DESIGN**

**4.1. GUI:**

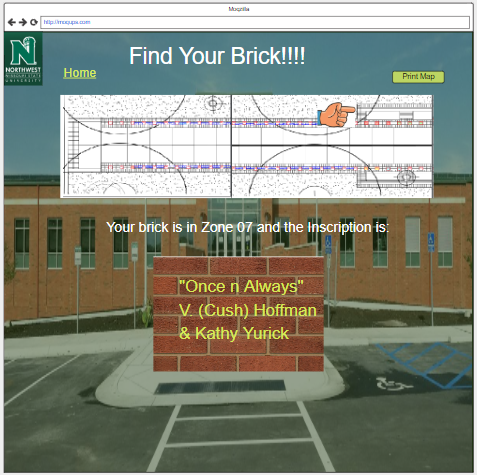
Mockups reflect the design choices for color schemes, layouts, typography, iconography, the visuals of navigation, and the overall atmosphere of the product.

**Web Site:**

Search Screen:

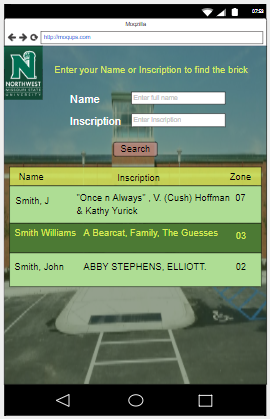
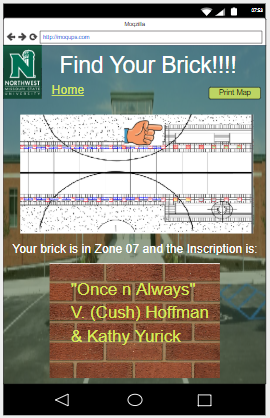


Map Screen:



**Mobile site**

Search screen: Map screen:

**UNIT 5. Technical Manual**

**5.1 Search page**

This page has Donor name or inscription text field. When the donor enters name or inscription it displays the search results in a table.

* We have UI validations for donor name and will display proper feedback about the count of the search results.

**5.2 Map Page**

Once the user clicks on any of the search result they should be able to see the map image with the corresponding zone numbers, and a brick/paver with inscription on it.

* We used showImage() function to display the map on the search page by hiding the table.

**5.3 Print Functionality**

If the user wants to take a print of the map image, they can print it by clicking the print button or they can even save the page as a PDF.

We used window.print() function that prints the whole page with the map and a brick/paver with inscription on it.

**5.4 Database Functionality**

We did not use any database for this project. Everything is made client side. The client can manage the excel sheet anytime without any issues.

**5.5 Security Issues**

There are no security issues with this project as we are not using any personal information of the donors. Client can use any public repo to store the excel sheet.

**UNIT – 6 END USER MANUAL**

**6.1 Search page**

**6.2 Map page**

**6.3 Print Functionality**