Joanna Lovell

Project Documentation



# Executive Summary

This summary can be taken directly from the project design document. Proofread and edit the section so that it is appropriate for the final project documentation.

This project will solve issues pertaining to poor phonebook management. This project solves these issues by creating a program that will house phone numbers, emails, and fax numbers on a web browser to allow people to look up phone numbers or other information like email or fax numbers. This eliminates the need for paper copies of a phonebook directory for everyone within the company. It also stops having to distribute new phonebook directories via paper copies when needing to update the directory because everything will be stored in the program and updated via the program.

A phonebook management system will benefit the company in many ways. One way is through communication within the company. This management system allows employees to look up phone numbers and email addresses quickly within the system. It will keep contacts in one place instead of on paper and spread in many different areas. Allowing one place of copy also allows updating the information faster and easier to do so. It's accessible anywhere; if an employee is on the go, they can open this directory whether they are working from home or some remote location. It is secure, so when an employee leaves the company, they lose all access to the management system.

People that will be able to use this system are people within the company. This is a company phonebook management system, so only people within a company will have access to this information. This system will allow people from within the company to look at a phone number up to call another person within the company.

This project is designed for a company to have a phonebook management system. This system allows users to add a new phone record, update phone records, delete phone records, search for phone records by the department or email, and display existing phone records. There will be a main phone number for the department. Then, there will be individual people in those departments with specific extension numbers. Each department has a unique email address, and everyone will have their particular email address as well. A department will have a fax number associated with them so that if someone needs to send a file to people electronically will be able to do so via fax

# Scenario | Program Usage Description

Use as many scenarios as needed to provide an overview of the system. The scenarios should give a screen shot and describe how the screen is to be used.

Graphical user interface, text, application

Description automatically generated

The register page is where a user will be able to register for the site so that they can login. They will put in their username, password, department, first name, and last name. Afterwards to register they will click the register button and to exit the page they will click cancel.

Graphical user interface, application

Description automatically generated

The login page is where the user will login to the system with their username and password and click the login button. They can cancel their login with the cancel button if needed. Through the login page the user can also get to the registration page by clicking the register button.

A picture containing text, screenshot, indoor

Description automatically generated

The search page is where users that are signed into the system are able to search for other people, departments, phone numbers, or fax numbers that are listed in the systems databases. The enter button will send the user to the page to display the information they are looking for and the clear button will clear the information out of the text box for the user.

Graphical user interface, text, application, table, Excel

Description automatically generated

Graphical user interface, text, application, table, Excel

Description automatically generated

Graphical user interface, text, application, table, Excel

Description automatically generated

Graphical user interface, text, application, table, Excel

Description automatically generated

The following four screenshots above are of the phone page. This page displays the information that the user asked to see.

# Reports

Provide an introduction to the reporting features of the system if there are any. At a minimum, there should be a listing of reports along with a description of the information the report provides. Students may also wish to add screen shots of reports to enhance the description of the reports.

# 

This report shows a list of people that are in the database. The report shows the department that they are attached to the email department they are attached to. The phone number and their extension number they can be reached at and the fax number that people can fax information to.

# System Architecture

Describe and diagram the system architecture. This section is set aside to document where code lives and runs. A graphical map of a live system works nicely here. Additionally, if it was necessary to employ any type of record-locking scheme, this is the place to discuss that.

The user will use a device to access the web applications content. The device will collect and display results to a user. The device also requests and responds to the content of the web application. The web server will gain the file system and database information.

Diagram

Description automatically generated

## Source Code Structure

Source code structure introduction. The following is a summary of the source code directories and their contents:

|  |  |
| --- | --- |
| **Code Directory** | |
| **Directory** | **Usage** |
| /Css | This folder provides all the code for Cascading Style Sheet code. This is for the style of the pages. |
| /Pages | This folder provides all the code for ASP.net code. These are the html pages such as the login, register, search, and phone pages. Along with the C# files. |
| /Foundation.js | This is a JavaScript file that contains JavaScript for the web application in different pages. |
|  |  |
| Phonebook.mbd | This is the database that the web application utilizes. |
| *Highlighted rows indicate directories containing source code.* | |

# Executables

Describe the executables that are a part of this project. If there are multiple executables break out each one into its own heading and give its name and a description of what function it plays in the system.

### PhoneApp (PhoneApp.exe)

This is the main folder for the application it houses all the information of the application. All its source code is stored here.

### EXE (EXEFileName.exe)

Description of the EXE and its usage in the project.

# Code Architecture

Explain how the system is put together. Provide an introduction and then dig into the database or data store design and then explore the internals of the code.

The system is put together through a database and through languages such as C#, JavaScript, and Asp.net. The database has four tables one for the user, department, email, and phone number. The user table stores information such as the first, last name, username, password, statusId, DepartmetnID, PhoneId, EmailId. The phonenumbertable which displays phone numbers and extension numbers is attached to the user table. The department table which department name and fax number is attached to the user table. The email table which displays email addresses is attached the user table. The asp.net, C#, code has a login menu a registration menu so users can login and register from a company. Then a search page so a user can search for people through the database and display that information on the phone page. On the JavaScript page there are functions such as a cancel function that allow a user to cancel what they are doing in the web application and functions to set the background color of a web page.

## Database or Data Store

Introduction to the database or data store.

A picture containing graphical user interface

Description automatically generated

Additional information about the database design should go here.

## Views, Stored Procedures and User Defined Functions

If views, stored procedures and/or user defined functions are used they should be described here.

There are not any stored procedures or user defined functions in my program.

External Files & Data

If any external files are used and/or if information is stored in the Registry, that information should be documented here. If this information needs security of some type that should be documented here as well. Be sure to address WHAT security steps have been taken if they are required.

Phonebook.mbd is an external file that is used it’s a Microsoft Database file.

Programming Language | C#.NET

Provide details about what language the project is written in. – Be sure to identify any components, references and/or external DLLS.

The programming languages utilized in the program are C#, Asp.net, Css, and JavaScript.

Project Classes

Classes within the project are used to abstract re-usable pieces of code. Classes are also used to group related values, known as properties. The project utilizes these classes:

### JavaScript Cancel| cancel()

The class is shared among the login and search pages that when a person clicks cancel the information will be erased out of the text boxes.

### Asp.net Gridview| NameSearch

In the Asp.net page it utilizes one grid view instead of using three or four different gridviews to show information in a different way. It does this through backended C# code.

### JavaScript ColorChange| ColorChange()

Changes the background color to gainsboro on the login page and the search page.

### 

Project Modules

Modules are used for procedural based code that does not require state data like class modules do. Complete the introduction to modules.

I do not have any program modules in my program application.

### Short Module Description | ModuleFileName.cls

Class description here.

### Short Module2 Description | ModuleFileName2.cls

Class description here.

### Short Module3 Description | ModuleFileName3.cls

Class description here.

Program Start and End Flow

Describe and then diagram the program flow. Here is an example of program flow from a fat-client based PC application.

The user will be promoted to the login page. They either have an account or don’t have an account. If they have an account, they will login. If they don’t have an account, they will go to the registration page if they don’t want to register the program will be terminated. Once a user has an account and they login they will be sent to the search page. On the search page they will type in the text boxes what they are searching for. Then, they will be sent to the phone page where the database information will be presented and then the program terminates.

Diagram

Description automatically generated

Summary

Briefly summarize the system documentation here. This section should be no longer than 1/3-1/2 of the page.

This project is designed for a company to have a phonebook management system. This system allows specific users to add a new phone record, update phone records, delete phone records. A user can search for phone records by the person’s name, department, email, phone, or fax number. There is a main phone number for each department with each person in the department having extension numbers to that phone number. Each department has a unique email address. A department will have a fax number associated with them so that if someone needs to send a file to people electronically will be able to do so via fax.

### This documentation has concluded the executive summary, scenarios, the system design, system architecture, build and release process, flow, client installation, and developers setup instructions.

# APPENDIX B (BUILD AND RELEASE PROCESS)

Describe the build & release process required to implement an update.

This section will describe all the stages that are involved in a release from development and testing to deployment. Release management is required when there are changes to a new product or an existing product. There are a couple different important steps when it comes to release management these are planning the release, building the release, testing the release, preparing the release, deploying the release, and maintaining the release. We often use the SDLC (Software Development Life Cycle) to plan and initiate our releases. The SDLC encompasses planning, gathering requirements, designing, software development, testing, deployment, and operations and maintenance.

Planning the release is an important step because this step will gather the foundation of what the release is from start to finish. During the planning phase a scope of the overall project will be built, timelines, delivery dates, and requirements will be gathered. Some will put these in a chart or a checklist but, preferably anything in this project will be put in a workflow chart. Using this type of flowchart will allow a visual of the actual tasks and deadlines and they can be done in different colors if needed and set at different priority levels. After the chart and checklist are done then it needs to be reviewed to make sure that everything is there.

**Testing Planning**

Once the planning phase has been approved, then the system design and building process can begin. Once the program is then built it will need to go through testing. Testing is going to vary depending on the release some releases will need more extensive testing than others. One of the main testing that can be done to all testing scenarios no matter what is User Acceptance Testing this is where the client will get a feel of the system and make sure that they like the design and if they want any changes then they can request those changes and have the system sent back to the development phase until they feel the system is where they think it should be. Documentation of all bugs will need to be made so that its been aware and fixed in the program.

**Deployment Planning**

After all the bugs and fixes have been made, then the project will go on to the deployment phase. This phase is where the product becomes live. This stage will also message and educate on the product at the end user and company. All users of the system will be notified of changes to the release and how they need to operate the new feature. Training may be required depending on the extensive or non-extensive release. The performance of the program will be monitored in this stage and to monitor any problems that could be lingering and get them identified and documented. Documentation of all bugs will need to be made so that it’s been aware and fixed in the program.

Documentation of any updates or changes to the system will be made and at the end of the project will be filed under the completed portion but filed in records.

**Maintenance planning**

Maintaining a system is just as important as monthly upkeeping of cars. It’s an important part of the process to keep a system maintained and stable. Software is not always perfect and will need fixes and updates to the system to maintain optimal performance. Maintenance is to correct the faults in programs, improve performance and or other attributes, and modify the environment of the product. Maintenance on the program will be done quarterly to ensure the program is working correctly and efficiently also as changes are made maintenance will be done. Documentation will be done to document maintenance that was established and done.

# APPENDIX C (CLIENT INSTALLATION INSTRUCTIONS)

Detail how a client machine or device is prepared to utilize the project.

A client machine or device would need to have the files installed onto their device or machine. Then the client would need to choose which would be best for them if they would like to host the web application. If hostable then they can simply go to the web page and login with credentials given. If not hostable, another route would need to be taken to get the application on a platform so that it could be opened without the need of a compiler.

# APPENDIX D (DEVELOPER SETUP INSTRUCTIONS)

Detail how a developer must setup their environment in order to work on the code.

A developer would need to have a IDE installed on the computer to run this program. The IDE would need to be able to run asp.net, Css, C#, and JavaScript. Any compiler like Visual Studio code or Visual Studio 2019/2017 would work on this program. From there a developer would need to use Microsoft Access or a database of their choice with modifications to the code.