Software Design Description

Version 1.0
<<Annotated Version>>

May 30, 2018

Program Showcase

Richard Carroll
Zachary Cash
Melissa Gonzalez

Submitted in partial fulfillment
Of the requirements of
CS 492 Senior Project

Table of Contents

List of Figures

1.0. Introduction

1.1. Purpose

This document talks about how our project, Program Showcase, operates. It will include how we implemented each feature described in the SRS and will go into exact detail on what each module of code does and how our database scheme operates.

This document will be catered to software developer audiences.

1.2. Scope

This project is separated into two parts. This first part is the front-end which handles how users interact with the site overall. The second part is the back-end which contains the database and a server that waits for incoming requests from the front-end and returns results back to the front-end.

1.3. Glossary

Term	Definition
Emulator	Obtaining to the screen that will host the output of an executing program.
JQConsole	This refers to a widget technology that provides an in-browser console.
Program/s	A set of one or more Python files that produce and output when executed.
Programmer	Those creating an account and uploading programs.
General user	Those searching or executing programs.

1.4. References

Vision and Scope Document

Revised SRS Document

1.5. Overview of Document

- Chapter 2 is a Deployment Diagram that shows the flow of information and how programs get executed.
- Chapter 3 is the Architectural Design. This is presented in the form of a diagram.
- Chapter 3 is the basic Data Structure Design for the website.
- Chapter 4 is on User Interface Design. The expectation of what a user is to expect from the interface is detailed here.
- Chapter 5 describes the minimal real-time design for this project.
- Chapter 6 describes the minimal structure of the Help System.
- Chapter 7 exhibits the Use Case Realizations described in our SRS.

2.0. Deployment Diagram

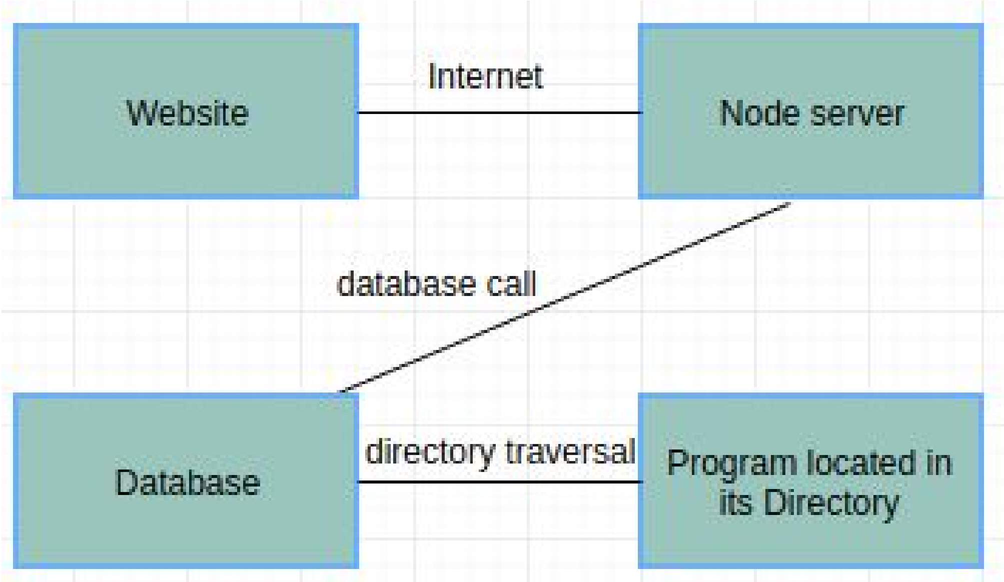

Figure 1 - Deployment Diagram

A programmer or general user will connect to our website via their Web browser (which we are not in control of). After accessing our website, both programmers and general users will have the option to sign in, create an account or search projects. Signing in requires a username and password which gets checked against our database to be authenticated. Creating an account requires a username, email, and password. This information gets stored in the database and the passwords are hashed using bcrypt. Searching for users or programs queries the database for username and if it finds the username it will take the searcher to the users page.

3.0. Architectural Design

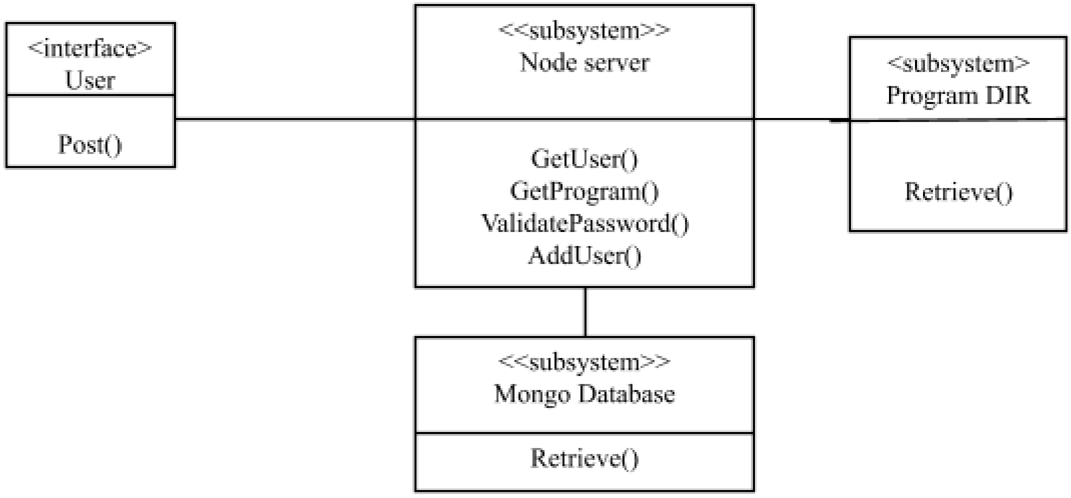

Figure 2 - Top-Level Architectural Diagram

User represents either a programmer or general user. The User Interfaces are covered in

Chapter 4.

4.0. User Interface Design

4.1 Program Showcase Interface

The interface for Program Showcase will feature the basic options that both programmers and general users have on the main page along with the title and description of the site's purpose above them. The three options will be divided into sections on the main page which will be labeled (from left to right) “Welcome”, “Returning User”, and “Projects” which correspond to “create account”, “log in”, and “search” options.

When either “create account” or “log in” option is chosen, a new page will open with a form for the user to fill in. If there is missing information or passwords don't match, a small tooltip or popup will alert them of the specific information that is missing or incorrect. When the “search” option is chosen, a new page will load. It will contain a search bar with three buttons underneath labeled (from left to right) “Beginner”, “Intermediate”, and “Advanced”. A general user will have the option of searching for a specific username or selecting programs by levels.

Regardless of how a program is chosen, it will link a general user to the programmer's public profile page. Here, they will see a list of programs the programmer has uploaded to share. Each program listed will be a link that redirects them to an emulator. Within the emulator, the program's output they chose to execute will appear. If the program requires input, the emulator will allow the general user to interact with it in order to provide the input needed.

5.0. Real-Time Design

There are only a few real-time design considerations within this project.

The programmer is the main user of the site. They have additional options other than searching for other projects such as signing in and uploading.

Program Showcase is designed for multiple users but the concurrent usage is handled on the client-side. That is, the main page will execute on the user (client) side and it will then make requests on Program Showcase's server. These requests are handled by the server. Errors such as incorrect information, the page doesn't exist, or broken links will be handed in the front-end in order for the user to see and take action.

6.0 Help System Design

The help design for this project is very minimal.

The site for Program Showcase will have a Help page that all users can reference at any time if needed.

The Help page will contain a list of possible error messages that a user may encounter along with suggestions to avoid or overcome any obstacles within the site.

7.0. Use Case Realizations

	3.0	4.1	5.0
When a programmer logs in they are sent to their personal private page	X	X	
When a general user runs a program	X	X	X
When a programmer uploads a program		X	X
Programs will be able to be queried by their difficulty		X	

7.1a User Case for a Programmer

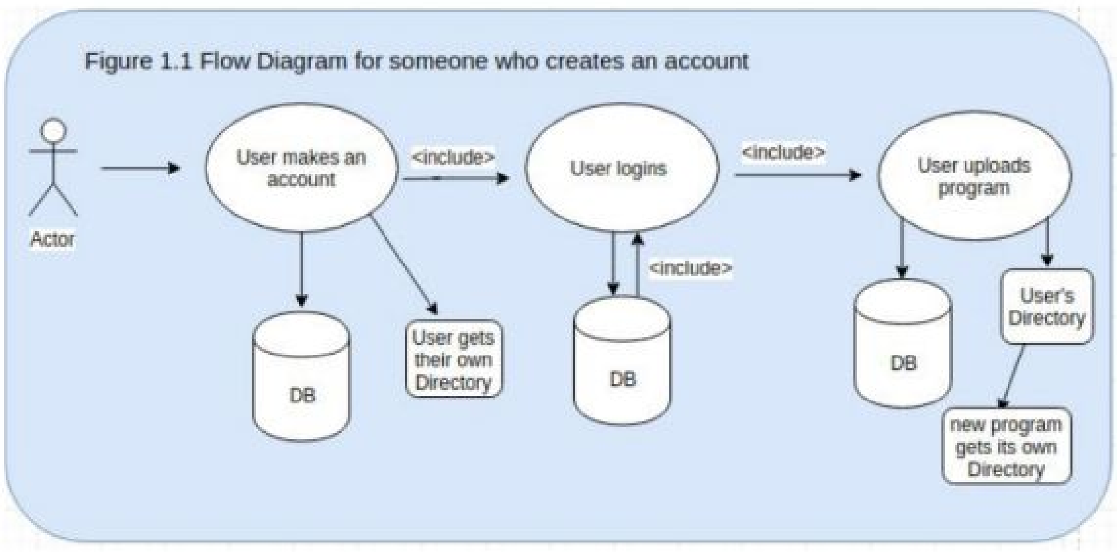

Figure 1.1 Flow Diagram for someone who creates an account

This diagram describes a use case for a user creating an account. First, the user inputs information into website that gets sent to the database for future validation purposes. There is then a directory created with the title of their username which is where there will be another directory that includes their project name and inside of that their source files. There is also other information about the files that are stored in the database such as number of arguments, their type, and a description of what the project does.

7.1b Flow of Storing Programmer Info

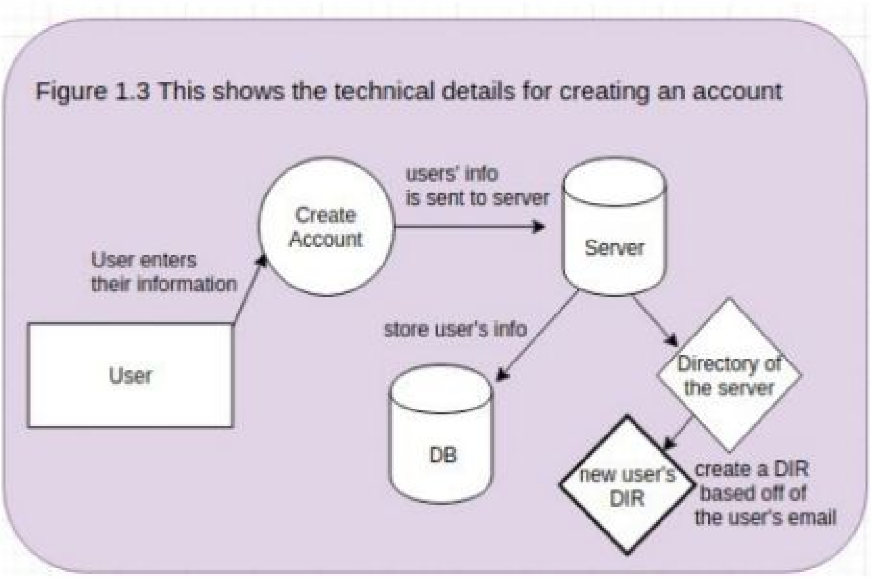

Figure 1.3 This shows the technical details for creating an account

User
User enters their information
Create Account
users' info is sent to server
Server
store user's info
DB
Directory of the server
new user's DIR
create a DIR based off of the user's email

This diagram is an example of the flow of data when a user creates an account. When the user enters the site and inputs a username, unique email, and password, they will be able to successfully create an account. This information is required to create an account and is then sent to the server to then be stored in the database. Because a user's account is where they will the have the option of uploading files and viewing ones they have previously uploaded, a directory for this user will be created. Within this directory is where the files will be stored.

Index