SE 3XA3: Software Requirements Spann

Team 5
Christopher Stokes — stokescd
Varun Hooda — hoodav

October 11, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Oct. 1, 2016	1.0	Initial Changes

This document describes the requirements for Spann. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (**?**). If you make further modifications to the template, you should explicity state what modifications were made.

1 Project Drivers

1.1 The Purpose of the Project

The purpose of the project is to develop a web browser based Python IDE application. The application will provide an environment similar to desktop based integrated development environments but with the convenience of a seamless experience regardless of their operating system or hardware platform with the only requirements for the user is a modern web browser.

1.2 The Stakeholders

1.2.1 The Client

The client for whom this application is being developed is Dr. Smith, the professor of Software Engineering 3XA3.

1.2.2 The Customers

The customers of the application will be python developers looking for a continent platform that allows them to develop from almost anywhere, on almost anything with a internet connection.

1.2.3 Other Stakeholders

1.3 Mandated Constraints

The project needs to run on software and hardware that McMaster University has and has the licenses for.

1.4 Naming Conventions and Terminology

1.5 Relevant Facts and Assumptions

The application will assume the user has a modern, HTML5 compatible browser that has JavaScript enabled. The application will be tested to ensure it is functional on the major modern browsers(insert reference to list of browsers in appendix here).

User characteristics should go under assumptions.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

2.1.2 Work Partitioning

In general the split of work is approximately equal in client and server side code. The difference in who is responsible for the underlining architecture development and who is responsible for the application specific development. In general the underlining setup and architecture development is handled by Christopher Stokes and the application specific development by Varun Hooda.

Christopher Stokes:

- Project and code design
- Database design
- SQL code generation framework
- UI framework design
- UI design
- Testing

Varun Hooda:

- Database design

- UI design and development
- API development
- Client and server side algorithm design and development
- Testing

2.1.3 Individual Product Use Cases

Algorithm Testing: A main use case of this project is to be able to design individual parts of larger projects or algorithms without it interacting with the larger project. Because of how quick this project is a creating projects it makes it incredibly easy to test small sections of code during development. This can be done in a project or in the console.

Full Project Development: This project can be used to develop full projects with virtual know limitations on what can be supported. There are a number of reason this would be done, such as it means the developer does not need to create a local environment and is able to develop on any deceive with access to the web.

2.2 Functional Requirements

3 Non-functional Requirements

3.1 Look and Feel Requirements

This application needs to maintain a unified enterprise look and feel. A major goal of this application is to increase development speed, this means it is incredibly important that the user experience is clean. To achieve this, options such as save, properties, and right-click must be constant throughout the application. As a target, this application must respond and feel like a desktop application while running in browser meaning all actions that one would expect to work in a desktop IDE such as Eclipse must work in browser.

3.2 Usability and Humanity Requirements

Users of this project are generically of a technical background and the nature of the work is technical. This means the product is able to be designed in a more technical nature but it is still import that the product is usable by ensure a consistent design and button location across all screens.

3.3 Performance Requirements

3.3.1 Client

The client side UI must be quick and responsive as well as not using a lot of resources as it runs in the browser. It is most important that the application can run asynchronously at all times to not lock up the browser.

3.3.2 Server

For the server it is very important that it is fast and is able to handle a lot of requests. It also needs to run its processes in a proper asynchronous manner as the processes are long running.

3.4 Operational and Environmental Requirements

3.5 Maintainability and Support Requirements

It is highly important that the source for this project is maintainable as the features of the IDE will grow as more advanced tools are added to encompass or use cases and parts of the development processes. Furthermore web APIs and support changes incredibly fast so it is important that the code stays up to date to support the best technology.

3.6 Security Requirements

This application should provide a secure platform and carry out its functionality in a secure manner. This means the application needs to meet the following a set of security requirements:

- The application cannot allow anyone except the owner of the account to view/modify to the files associated with the account and the settings associated with the account.

- The application does not allow anyone to intercept the data while it is being transferred over the network.
- The application executes user code in secure manner, isolated from the rest of the system to ensure any malicious code doesn't compromise the security of the system and application.

3.7 Cultural Requirements

¡¡¡¡¡¡¡ Updated upstream ??????? ======= ??????? ¿¿¿¿¿¿¿ Stashed changes The project may need to be translated or support translation if a significant number of users' primary language is not English.

3.8 Legal Requirements

3.9 Health and Safety Requirements

¡¡¡¡¡¡¡ Updated upstream ??????? ======== ??????? ¿¿¿¿¿¿¿ Stashed changes

This section is not in the original Volere template, but health and safety are issues that should be considered for ev-

ery engineering project.

4 Project Issues

4.1 Open Issues

4.2 Off-the-Shelf Solutions

The are various off-the-shelf solutions available for writing and running code on the web browser, but these solutions have little to no support for the python programming language besides being able to provide simple code completion and ability to execute the code. Thus, for this project, we're aiming to both support execution of python, code completion and add IDE-like, developer friendly features (fill in some IDE-like features we'll have)

4.3 New Problems

4.4 Tasks

4.5 Migration to the New Product

4.6 Risks

One inherent risk with allowing users to execute code on your servers is the user's ability to perform malicious actions. This can result in damage to the hardware, the software stack and to the data on the server. Another risk is the possibility of some fault in the system causing user's to lose data or the project to lose business critical data or damage the hardware or software stack.

4.7 Costs

The project will be mainly using free and open source (libre) software that is available without costs, as well as non-libre software this is available to us without cost. If the platform is to be scaled for public usage, the project will need to be hosted on some server (or multiple server depending on user adoption) which would have a regular cost.

4.8 User Documentation and Training

The project will be fully documented, including design documents, testing documents, well commented code.....

4.9 Waiting Room

4.10 Ideas for Solutions

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.