Software Requirements Specification

for

<Science & Tech Fest>

Version 1.0 approved

Prepared by <Kasiet Ryspaeva>

<Science & Tech Fest>

<10/09/2019>

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| W1 | 17/09/19 | Demo version released |  |
| W1 | 19/09/19 | Update design |  |

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1
   1. Purpose 1
   2. Document Conventions 1
   3. Product Scope 1
   4. References 1
2. Overall Description 1
   1. Product Perspective 1
   2. Product Functions 2
   3. User Classes and Characteristics. Speakers: All Speakers should have a registration at least a 5 days before and highlight his/her status as “Speaker” . Participants: No matter of age, nationality and status can apply at least a day before of this Fest. 2
   4. Operating Environment 2
   5. Design and Implementation Constraints 2
   6. User Documentation 2
   7. Assumptions and Dependencies 2
3. External Interface Requirements 3
   1. User Interfaces 3
   2. Hardware Interfaces 3
   3. Software Interfaces 3
   4. Communications Interfaces 3
4. System Features 3
   1. System Feature 1 3
   2. System Feature 2 (and so on) 4
5. Other Nonfunctional Requirements 4
   1. Performance Requirements 4
   2. Safety Requirements 4
   3. Security Requirements 5
   4. Software Quality Attributes 5
   5. Business Rules 5
6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

# Introduction

## Purpose

*The purpose of this project is to inform people about great fest “Science and Tech Fest”. In order to save time, money, resources, avoid false information, bots and malware*.

## Document Conventions

Font of Bold text is Times (Shows the topic of each one) size is 14 ; for text is Arial and size is 11. Document consist of 6 pages, 1st page is title page, 2nd page is overview page.

**Intended Audience and Reading Suggestions**

Reader that the document is intended for, are developers, project managers, users, testers, and documentation writers. We highly suggest to read the document, beginning with the overview sections and proceeding through each section and read carefully.

## Product Scope

Project goals:

1. Share information about the upcoming festival

Project objectives:

1. Agile

This project presents information about Tech and Science Festival. Project made in the form of a web-site. The site is put on the Internet using GitHub Pages. Website is open source project. It is available for all devices.

## References

This document and demo version are stored in the web browser:

https://techsciencefest.github.io/

# Overall Description

## Product Perspective

This site presents information about Tech and Science Festival. Project made in the form of a web-site. The site is put on the Internet using GitHub Pages. Website is open source project. It is available for all devices.

## Product Functions

1. Single-page site
2. 5 sections
   1. About festival
   2. Schedule
   3. Speakers
   4. Sponsors
   5. Contacts
3. Registration ability
4. Google map which show location of festival

## User Classes and Characteristics. Speakers: All Speakers should have a registration at least a 5 days before and highlight his/her status as “Speaker” . Participants: No matter of age, nationality and status can apply at least a day before of this Fest.

## Operating Environment

1. The software runs on PCs, laptops, mobile phones, all types of OS and phone types.

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

1. Get daily report from each team member
2. Checking Git repository (including comments and issues)
3. Communicate in Telegram group which is used specially for this project

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

The project is making for study and practice purposes.

Amount of required money 0

Amount of spent money 0

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

Project Manager: The project manager is responsible for creating the project plan (with input from those doing the work), managing risks, running the weekly team meeting and providing monthly status reports to senior management.

Documentation: The documentation manager is responsible for writing and editing documentation of website project.

Front-end developer: The front-end developer is responsible for writing of code and supporting project

QA:The QA is responsible for testing, running and debugging the results of the written code.

Designer:The designer is responsible for making visual effects on website.

Content maker: The content maker is responsible for making content of project.

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

Any acronyms and abbreviations have not used .

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>