
Software Requirements Specification

for

<The Best Vocational Test>

Version 1.0 approved

Prepared by <Gaston Martin, Camilo Petrignani Gorla>

<UNRC>

<date created>

Table of Contents

Revision History	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References.....	1
2. Overall Description.....	2
2.1 Product Perspective.....	2
2.2 Product Functions	2
2.3 User Classes and Characteristics.....	2
2.4 Operating Environment.....	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies.....	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces.....	3
4. System Features	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on).....	4
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
6. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	

1. Introduction

< taller de la materia de Análisis y diseño de sistemas de la UNRC. La Universidad solicitó hacer un sistema de cuestionario que se pueda utilizar en cualquier escenario Como ,por ejemplo, un procedimiento de orientación vocacional, en el cual se va a necesitar crear y mantener una base de datos en la cual se va a guardar las carreras disponibles, las preguntas a realizar en el cuestionario y los valores de las respuestas dadas por los usuarios y como influirán en la elección de la carrera >

1.1 Purpose

<Este programa esta destinado a todas las personas que no tengan seguro cual es la mejor carrera para ellos y para que los docentes puedan ver que carreras son las mas elegidas o con mas tendecia por los futuron nuevos ingresantes >

2. Overall Description

2.1 Product Perspective

<Se espera que el programa pueda ayudar a elegir las mejores carreras para los distintos usuarios que realizen el test y que sirva como base para nuevos programas que traten en la creacion de distintos test.>

2.2 Product Functions

<El Producto debe tener una interfaz de usuario interactiva para realizar el cuestionario de forma correcta. Debe dar una respuesta con las carreras sugeridas e información de las mismas, las cuales deben encontrarse disponibles en la UNRC. Debe proveer distintas respuestas para la misma pregunta. Una sección con todas las carreras disponibles, >

2.3 User Classes and Characteristics

<Para este producto solo tiene un usuario que es la persona que realiza el cuestionario>

2.4 Design and Implementation Constraints

<El programa esta limitado a solamente poder realizar un test vocacional, no crearlo o modificarlo, no se puede modificar la Base de Datos ya cargada en el sistema. El usuario no puede realizar el test sin dar su nombre, y no puede evitar ninguna pregunta, todas tienen que tener una unica respuesta.>

2.5 Assumptions and Dependencies

<Se asume que los usuarios que quieren realizar el test tienen conocimientos a la hora de usar un navegador web y la forma en que se tienen que responder las preguntas.>

3. System Features

3.1 System Feature

- ⑩ REQ-1 : COMO USUARIO QUIERO REALIZAR UN TEST Y SABER QUE CARRERA ES LA INDICADA PARA MI.
- ⑩ REQ-2 : COMO USUARIO QUIERO INGRESAR AL SISTEMA DE FORMA ANONIMA.
- ⑩ REQ-3: COMO ADMINISTRADOR QUIERO QUE EL SISTEMA DEVUELVA LA MEJOR CARRERA PARA EL USUARIO EN BASE A LAS RESPUESTAS DADAS.
- ⑩ REQ-4: COMO ADMINISTRADOR QUIERO PODER MODIFICAR EL CUESTIONARIO.
- ⑩ REQ-5: COMO ADMINISTRADOR QUIERO AGREGAR O ELIMINAR CARRERAS, RESPUESTAS Y PREGUNTAS.
- ⑩ REQ-6: COMO ADMINISTRADOR QUIERO SABER QUE CARRERA ES LA QUE MAS LE CORRESPONDE A LOS USUARIOS.

3.2 System Feature 2 (and so on)

4. Other Nonfunctional Requirements

4.1 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.>

4.2 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.>

4.3 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.>

4.4 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.>

4.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

5. 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

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

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.>