ID-ITS

RAnalyzer
Software Requirements Specification
For <Subsystem or Feature>

Version <1.0>

Revision History

Date	Version	Description	Author
04 Sept 2017	1.0	Initiation	Daniel Siahaan

Table of Contents

1. Introduction 4
   - 1.1 Purpose 4
   - 1.2 Scope 4
   - 1.3 Definitions, Acronyms and Abbreviations 4
   - 1.4 References 4
   - 1.5 Overview 4

2. Overall Description 4
   - 2.1 Scenario 4
   - 2.2 Context Diagram 5

3. Specific Requirements 5
   - 3.1 Functionality 5
   - 3.2 Non-Functionality 6

4. Supporting Information 6

Software Requirements Specification

1. Introduction

This section provide an overview of the entire document. It includes decription about the purpose, scope, definitions, acronyms, abbreviations, references and overview of the document.

1.1 Purpose

The purpose of this document is to describe the basic needs of end users that the system should provide and attributes, properties, constraints, and qualities tha the system should meet in order the system to achieve the goal of the system, i.e. to enable software engineer to analyze the dependency between requirements given the artifacts and deliverables of the project.

The document also describes the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

1.2 Scope

The software application should allow the end user, i.e. the software engineer, to create project and to check the dependencies within the software requirements, given the artifacts and deliverables of an existing project.

1.3 Definitions, Acronyms and Abbreviations

n.a.

1.4 References

UCS-CreateNewProject-ver-1.0	Use Case Specification Document of ‘Create a New Project’
UCS-OpenExistingProject-ver-1.0	Use Case Specification Document of ‘Open an Existing Project’

1.5 Overview

The rest of the document is organized as follows. The second section explain the overal description of the system to be implemented, i.e. the scenario and the context diagram of the system. The last setion describes the list of all functional and non-functional requirements.

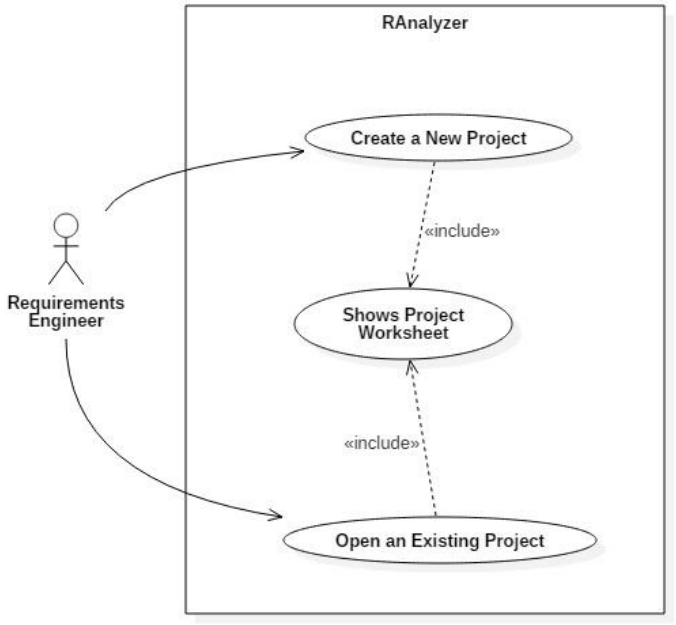
2. Overall Description

2.1 Scenario

Rivo is a requirements engineer working at ID-ITS. He is reponsible for analyzing the impact of requirements changes on other requirements during project iteration. Today, he has to come up with the result of their company on going project, i.e. SIDATA, an management information system for managing environmental parameter measurements on food and drinks. The application was ordered by the beaurau of environmental services, East Java Province.. It is a huge and long project. Change requests from end users immerged in every iteration which may cause domino-like changes on other requirements. The company needs to make a quick financial analysis in each new iteration in order to respond to the changes. Luckily, he has been using Ranalyzer to help him. He adds the necessary documents of the last working iteration on the existing project. Then, he executes the dependency analyzer module to triggered the process. Within a minutes, he get the result. He download the result, compose them into a document, and send it to the project leader from the email services. And he is done.

,

2.2 Context Diagram



ID	UC Name	UC Description
UC01	Create a New Project	Requirements Engineer can create a new project.
UC02	Open an Existing Project	Requirements Engineer can open an existing project

3. Specific Requirements

This section contains all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

3.1 Functionality

ID	Statement	UC	Priority
F01	Requirements Engineer can create a new project.	UC01	Must
F02	Requirements Engineer can open an existing project.	UC02	Must
F03	Requirements Engineer can add a design document.	UC01, UC02	Must
F04	Requirements Engineer can add a requirement statement.	UC01, UC02	Must
F05	Requirements Engineer can view requirements dependency of a project.	UC01, UC02	Must
F06	Requirements Engineer can check the dependency between requirements within a project.	UC01, UC02	Must
F07	Requirements Engineer can edit a requirement statement.	UC01, UC02	Must
F08	Requirements Engineer can delete a requirement statement.	UC01, UC02	Must

F09	Requirements Engineer can remove a document.	UC01, UC02	Must
F10	Requirements Engineer can view a list of requirement statements.	UC01, UC02	Must
F11	Requirements Engineer can save the project	UC01, UC02	Must
F12	Requirements Engineer can export the dependency graph into other format, i.e. jpg, bmp, and png	UC01, UC02	Optional
F13	The system shall be able to show a worksheet of a project	UC01, UC02	Must
F14	Requirements Engineer can add use case description.	UC01, UC02	Must

3.2 Non-Functionality

ID	Statement	Quality	Priority
NF01	Only authorized and authenticated user may use the application.	Security	Must
NF02	The system shall be able to accept the following design document: use case diagram.	Portability	Must
NF03	The design document is in XMI format	Operability	Must
NF04	The design document is in jpg format	Operability	Optional
NF05	The system shall be deployed as a desktop application	Portability	Must
NF06	The system shall open one project at a time.	Extendibiliy	Must
NF07	The project shall be stored in a file with a .ran extension.	Portability	Must
NF08	The project shall be stored in xml format.	Portability	Must
NF09	The default name for the file being saved is ‘Untitled’. If there is an existing file with the same name, a incrementaly number is added at the end, e.g. ‘Untitled1’, ‘Untitled2’, etc.	Operability	Must
NF10	Project worksheet contains two child windows, i.e. Statements List Window and UML Diagram Windows	Operability	Must
NF11	The system can only open a project file with a .ran extension that complies with the software	Operability	Must
NF12	The use case description should be written in XML format	Operability	Optional
NF13	The use case description should contains at least the followig information: basic flow, exceptions, alternatives, and extensions.	Operability	Optional

4. Supporting Information

n.a.