



Project Name: Yoshi Project Requirements

Version 2.0

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

Revision History

Date	Version	Description	Author
2014-11-10	1.0	Initial Draft	Martin Anev and Yuxing Chen
2014-11-25	2.0	“Yoshi Evolved” changes connected with the change of the requirements from Customer’s side	Martin Anev

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

Table of Contents

1. INTRODUCTION.....	3
1.1 PURPOSE OF THIS DOCUMENT	3
1.2 DOCUMENT ORGANIZATION	3
1.3 INTENDED AUDIENCE.....	3
1.4 DEFINITIONS AND ACRONYMS.....	3
1.4.1 <i>Definitions</i>	3
1.4.2 <i>Acronyms and abbreviations</i>	3
2. OVERALL DESCRIPTION	4
2.1 PRODUCT DESCRIPTION	4
2.1.1 <i>Interfaces</i>	4
2.2 HIGH LEVEL DESCRIPTION OF THE DOMAIN	4
2.2.1 <i>General functionalities</i>	4
2.2.2 <i>General limitations</i>	<i>Error! Bookmark not defined.</i>
3. GOALS AND REQUIREMENTS.....	5
3.1 GOALS	5
3.2 REQUIREMENTS.....	5
3.2.1 <i>Functional requirements</i>	5
3.2.2 <i>Priorities of the Functional Requirements</i>	5
4. USE CASES.....	6
4.1 ACTORS	6
4.1.1 <i>User</i>	6
4.2 USE CASES.....	6
4.2.1 <i>Specification of use case</i>	6
4.2.2 <i>Use case diagram</i>	6
4.2.3 <i>Description of use-case</i>	7
4.2.4 <i>Lists of Tables</i>	7
5. REFERENCES	10

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

1. Introduction

1.1 Purpose of this document

The purpose of this document is to introduce general functionalities of the project delivered by Team Yoshi in the Distributed Software Development done simultaneously in ‘Politecnico di Milano’ situated in Milan, Italy and ‘Mälardalen University’ situated in Västerås, Sweden.

1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction*
- Section 2, *Overall Description*, describes different interfaces and high-level description of the domain
- Section 3, *Goals and Requirements*
- Section 4, *Use cases*

1.3 Intended Audience

The intended audience is:

- The customer of the project
- The supervisors of the project
- Yoshi Team
- All related stakeholders
- Any developer with interest to continue or improve the project

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Community	Social unit of any size that shares common values.
Open Source Community	Community that develops open source software.

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
API	Application Programming Interface
G	Goal
FR	Functional Requirement

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

2. Overall description

This section describes the general factors that affect the software product and its requirements, providing background for specifying concrete requirements in the next sections of the document.

2.1 Product description

The software product is completely independent from other systems, making the software product self-contained.

2.1.1 Interfaces

- External interfaces - The software product does not provide any external interfaces.
- *User interfaces* - The software product should have user interface that is graphical and interactive. The user should be able to visually understand and recon characteristics of communities as well as visualize them. The user should be able to focus on one characteristic or inspect multiple communities and their visualization at the same time, in the same screen.
- *Hardware interfaces* - The software product does not provide any hardware interfaces.
- *Software interfaces*- *The software product should connect to Database Management System and Apache Server.*

2.2 High level description of the domain

The software product that will be delivered is Yoshi. Yoshi is an Eclipse Plugin ~~web application~~ intended for helping people to measure, compute and study social community types. For instance, it will allow users to measure social and organizational characteristics of open-source communities (e.g. structure, formality, informality, etc...) measured by different metrics (e.g. collaboration between community members, common projects, list of followers, collaborators, contributors, etc..).

2.2.1 General functionalities

Yoshi should provide general functionalities for managing:

- Connection
Yoshi should be able to connect to communities from which it should gather the data.
- Measuring
Yoshi should be able to measure metrics from the communities to which it connects
- Visualization
Yoshi should be able to outputs the measured data (e.g. values of the metrics, specifying what type is the observed community).

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

3. Goals and Requirements

This section describes the goals, functional and non-functional requirements that affect the software, providing background for specifying concrete use cases which are presented in the next part.

3.1 Goals

We are giving the general goals of the project following.

[G1] Evaluate fitness function of the performance metrics.

[G2] Allow users to understand the characteristics of the researched community

[G3] Allow users to visualize the characteristics of the researched community

[G4] Allow users to focus on one characteristic or inspect multiple communities and their visualization at the same time, in the same screen.

3.2 Requirements

3.2.1 Functional requirements

[FR1] The product should be able to compute the grasped data

[FR2] The product should be able to store the computed data

[FR3] The product should be able to visualize the stored data

[FR4] The product should be able to compare characteristics on multiple communities

[FR5] The product should be able to visualize the comparison of characteristic on multiple communities and at the same time, in the same screen

3.2.2 Priorities of the Functional Requirements

The Functional Requirements are assigned with different priority levels as follows:

A-High

B-Medium

C-Low

[FR1] The product should be able to compute the grasped data	A
[FR2] The product should be able to store the computed data	A
[FR3] The product should be able to visualize the stored data	B
[FR4] The product should be able to compare characteristics on multiple communities	C
[FR5] The product should be able to visualize the comparison of characteristic on multiple communities and at the same time, in the same screen	C

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

4. Use Cases

4.1 Actors

4.1.1 User

A person that has fully functions provided by the system such as select a community to analyse and report the relevant characteristics. *The registration is necessary and the system may record the habits (data) of the users.*

4.2 Use cases

4.2.1 Specification of use case

We also giving the priority to each use case as before:

- A – High
- B – Medium
- C - Low

Use cases:

1. visualize computed characteristics (A)
2. compare communities' characteristics (choose different communities with the same (common) characteristics) (B)

4.2.2 Use case diagram

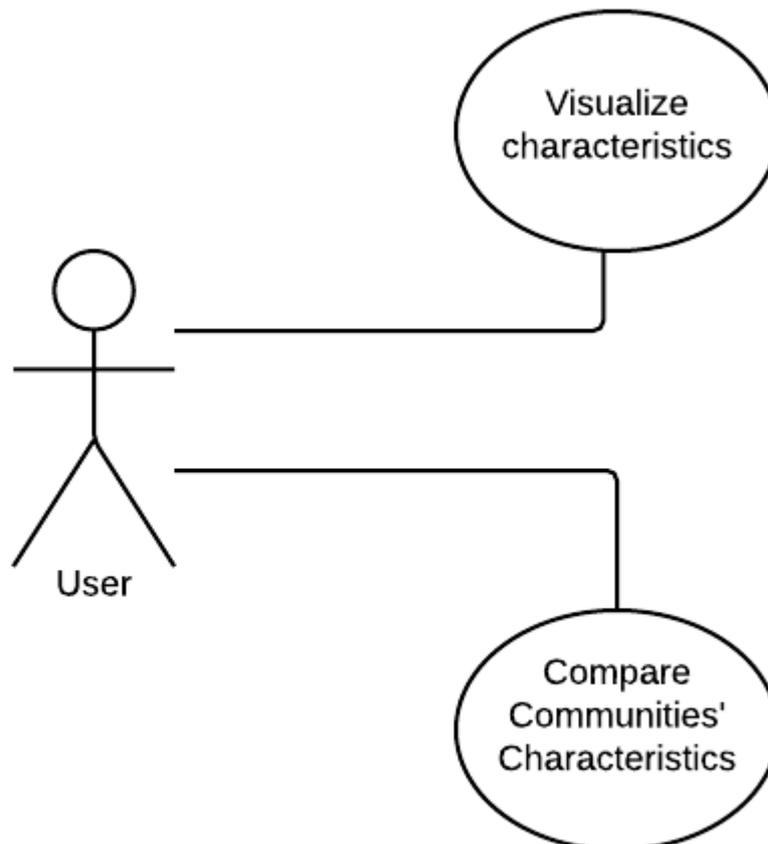


Figure 1 General Use Case Diagram

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

4.2.3 Description of use-case

In this section is discussed the detailed description with sequence diagram to the various use-cases related to the main use-case diagram defined above. For simplicity and readability, here are presented sequence diagram of the more complex use cases and the more simple ones are only described in textual form.

4.2.4 Lists of Tables

Table 1 Use Case: Visualize computed characteristics 7
Table 2 Use Case: Compare communities characteristics 9

Table 1 Use Case: Visualize computed characteristics

Name	Visualize computed characteristics
Actors	User
Entry conditions	After the system computed the system shows the result page
Flow of events	<ol style="list-style-type: none"> 1. The user starts the system 2. System provides choices for visualizing already computed results 3. The user selects the desired option <ol style="list-style-type: none"> 3.1 The user clicks “visualize the result” 4. The system visualizes the result
Exit conditions	The user quits the activity
Exceptions	No exception

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

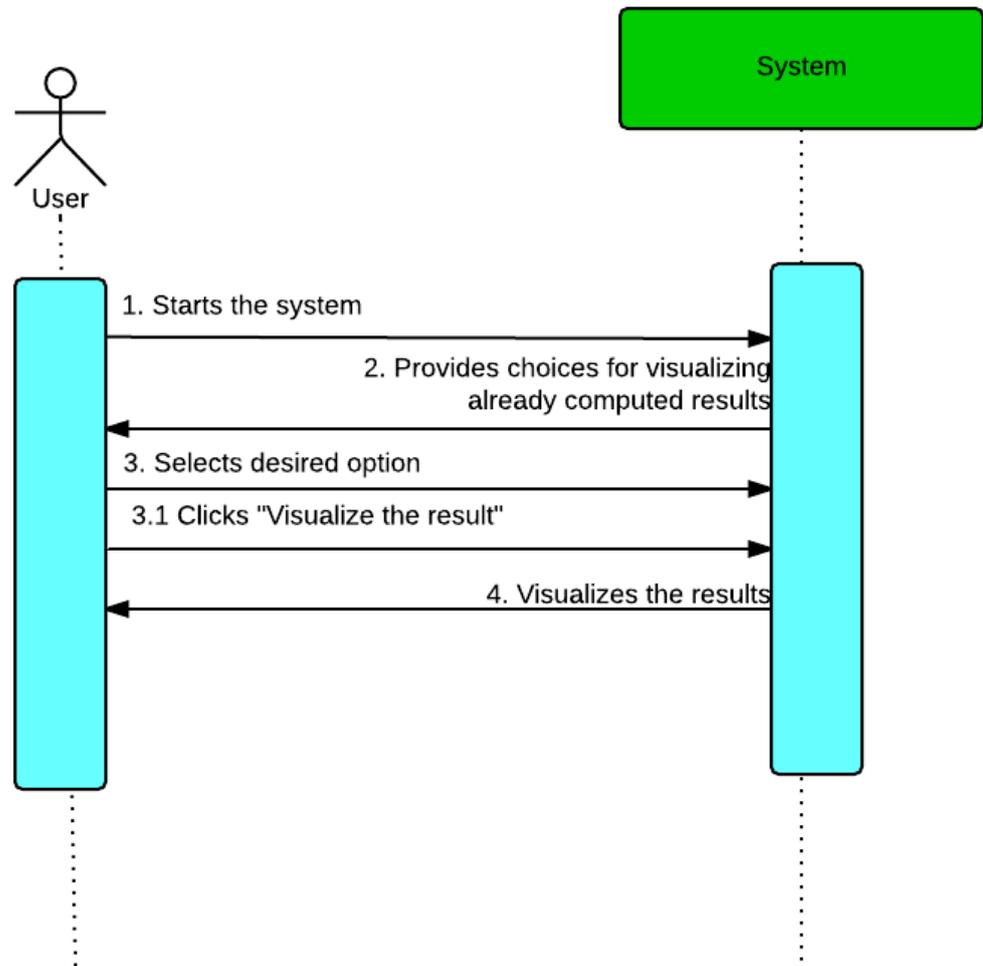


Figure 2 Sequence diagram: Visualize computed characteristics

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

Table 2 Use Case: Compare communities characteristics

Name	Compare communities characteristics
Actors	User
Entry conditions	The user has successfully runs the system and visualized characteristics of community
Flow of events	<ol style="list-style-type: none"> 1. The user clicks “Compare This Communities To Another” 2. The system provides choices from already computed communities 3. The user selects the desired option <ol style="list-style-type: none"> 3.1 The user clicks “Compare” 4. The system visualizes the results of the two communities
Exit conditions	The user quits the activity
Exceptions	No Exceptions

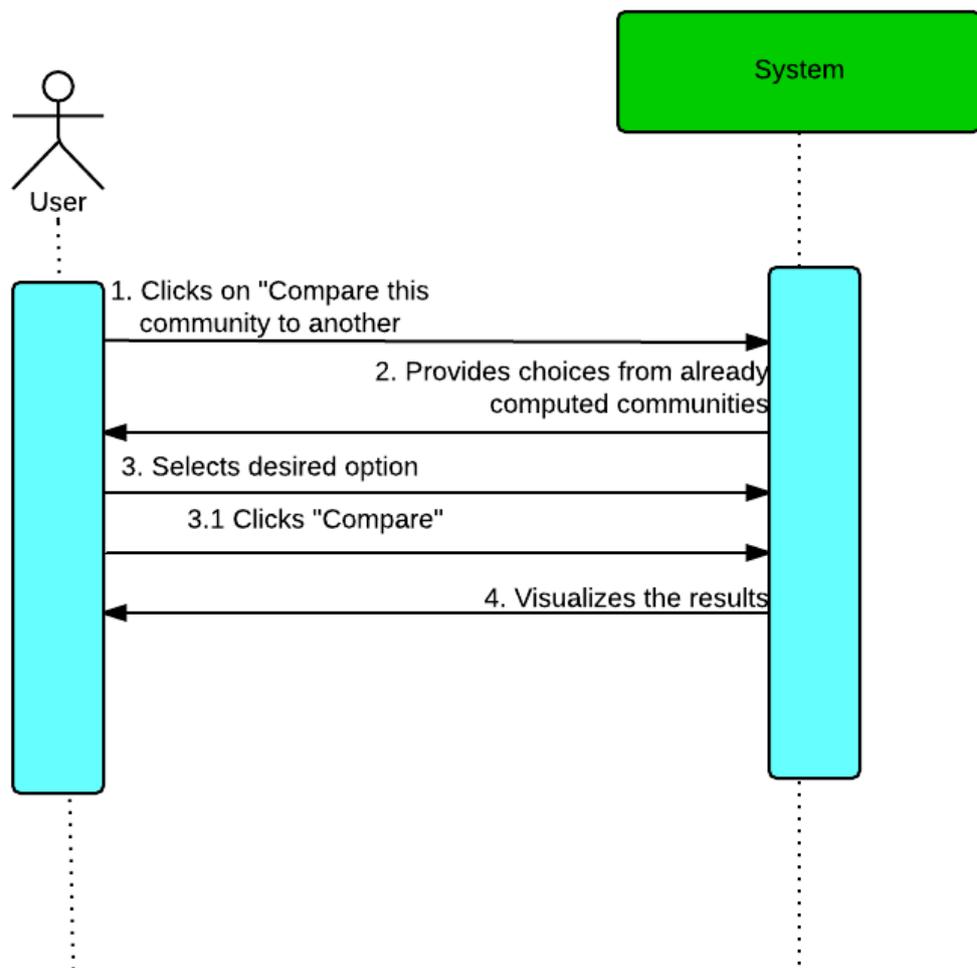


Figure 3 Sequence diagram Compare communities' characteristics

Yoshi	Version: 2.0
Project Requirements	Date: 2014-11-25

References

1. [IEEE Recommended Practice for Software Engineering Requirements Specifications](#)
2. [2] MetricsGrimoire - <https://metricsgrimoire.github.io/>
3. [3] VizGrimoire - <http://vizgrimoire.bitergia.org/>
4. [4] GPL v2 - <http://www.gnu.org/licenses/gpl-2.0.html>