Software Requirements Specification

for

Teamwork System

Version 1.0 approved

Prepared by Eagle

Software Development Workshop III

2020/3/8

Table of Contents

Table of Contents ii

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 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 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. System Features 3

3.1 System Feature 1 3

3.2 System Feature 2 (and so on) 4

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 4

4.4 Communications Interfaces 4

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 6

Appendix C: Issues List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Zhang Qiaomu, Tian Zhanming,  Zhang Haibin,  Li Jiajia | 2020/03/08 | Prepare SRS apart from section 3 & 4.1  Q.M. Zhang for section 1  Z.M. Tian for section 2  J.J. Li for section 4  H.B. Zhang for section 5 | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to capture the complete software requirements and preview some elements of the analysis model of the program Teamwork System (T.S.).

T.S. is a web application which can help teachers to form teams and then assess the contributions from team members during the process of task fulfillment in a course project. In detailed, it contains two sub-systems for two different kinds of users.

**Teacher system**: Teacher system helps teachers to initialize students’ accounts, manage submissions in the courses, forming teams and calculate the final assessment of the students.

**Student system**: Student system is used for students to form groups and manage the group such as the election of team-leaders and making an assessment for each other.

## Document Conventions

This document will be written using two different fonts, Arial for the main content. Time New Roman for varying levels of titles. Main content’s font size should be in 11pt. Bold fonts are used for emphasis or as headings. Each heading may contain subheadings in ascending order. Every requirement statement is assumed to have its own priority as to define in a most appropriate way the system behavior. Besides, various figures represent the described system, where it is needed, and serve only for a better understanding of the deployment.

## Intended Audience and Reading Suggestions

This document is intended for any users, developer and tester that needs to understand the basic system architecture and its specifications. Here are the potential uses for each one of the reader types:

* **Developer**: The developers who wanted to modify the system, must read the requirement and other features definition in order not to misunderstand the functions and do some reduplicative works.
* **User**: Users should be confirmed to understand the correct use of this system and some essential environment that is needed before using it. Then to find out whether the necessary functions exist, please check product features.
* **Tester**: Tester must make sure if the functions work well and if those features fit the requirements.

For each one of the reader types to better understand this document, here is a suggestion of the

chapters to read in this document:

* Developer:(1.1 ,1.3 ,2.2 ,2.3 ,2.5 ,2.7 ,3 ,4 ,5 and rest)
* User:(1, 2.2 ,2.3, 2.6, 4.1 ,5)
* Tester:(1 ,2.1 ,2.2 ,2.5 ,2.7 ,3 ,4 ,5)

**Overview**

1. **Introduction**: Provide an overview of the application, describe the document structure and point the individual objectives.
2. **Overall Description:** Provide the specification of the system model, the classes model, the main constraints and the list any assumed factors used within this document.
3. **System Features**: Provide an analysis of the requirements by feature.
4. **External Interface Requirements**: Provide the visualization of the program and the requirements that are related to hardware, software and networking.
5. **Other Nonfunctional Requirements**: Provide some additional constraints that apply to factors such as performance, safety and security.

## Project Scope

This software aims at helping teachers to calculate the final assessment by a given formula for students’ contribution. Besides, its automatically grouping function do assist teachers and students a lot. The benefits of this are that it saves time for teachers from assessing students and the traditional artificial grouping.

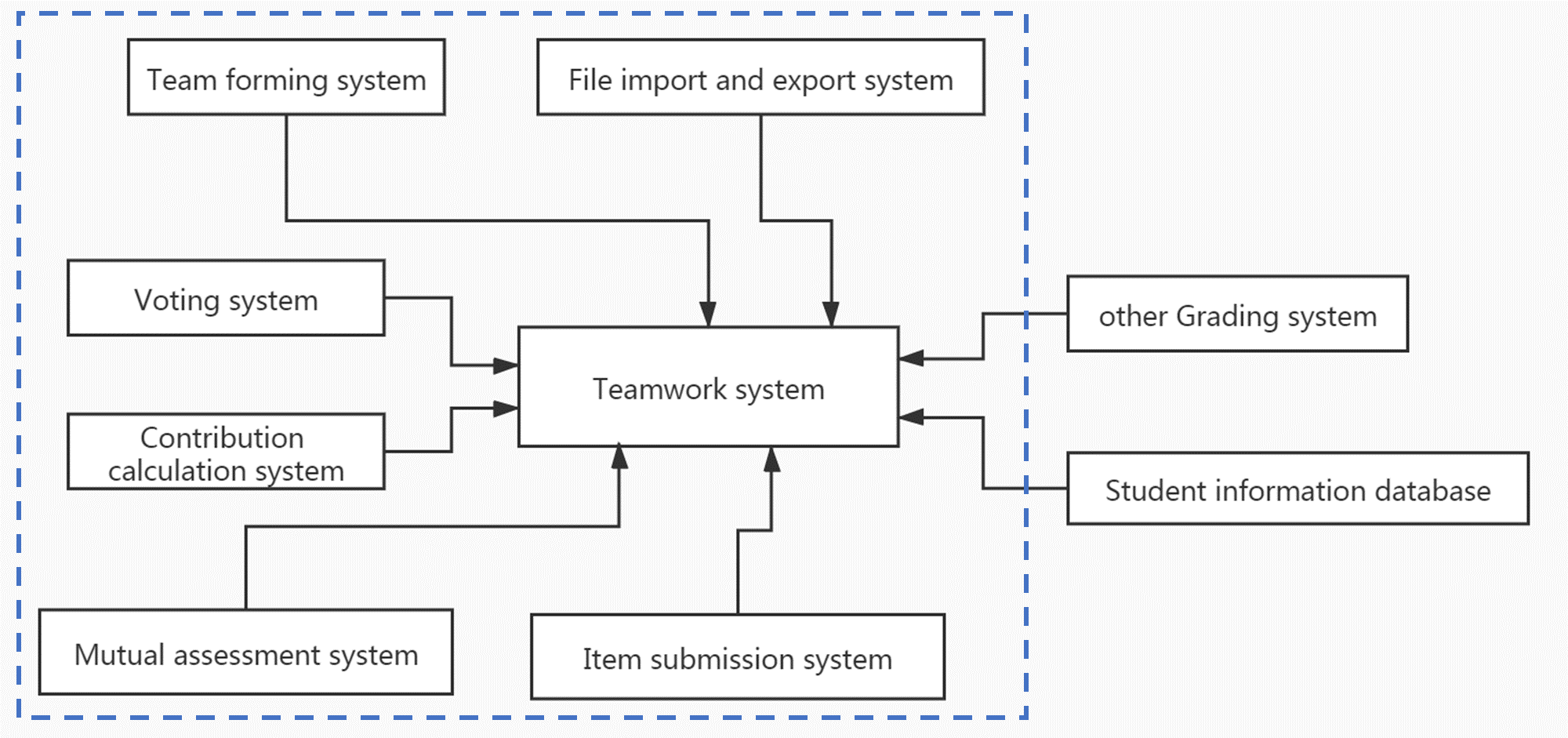
As the description above, the goal and objective of the software is set as an automatically-calculate assessment function. With it, teachers won’t need to do the calculation by themselves, which may even cause some calculation mistakes.

## References

By now, no document is needed to reference.

# Overall Description

## Product Perspective

The main components of the teamwork system include several sub-systems and connects to other systems. Outside the system, the teamwork system requires the student information like name and student I.D. from the Student information database. The exported contribution file will be used in another existing Grading system. Inside the system, there are systems providing various functions. Team forming system let students to form into groups. File import and export system enables teachers to import student information and export contribution file. Voting system enables students in a group to elect a group leader. Contribution calculation system enables individuals in a group calculated by algorithms. Mutual assessment system enables students to evaluate each other in a team. Item submission system allows teacher to upload item into the teamwork system.

Context model of Teamwork System

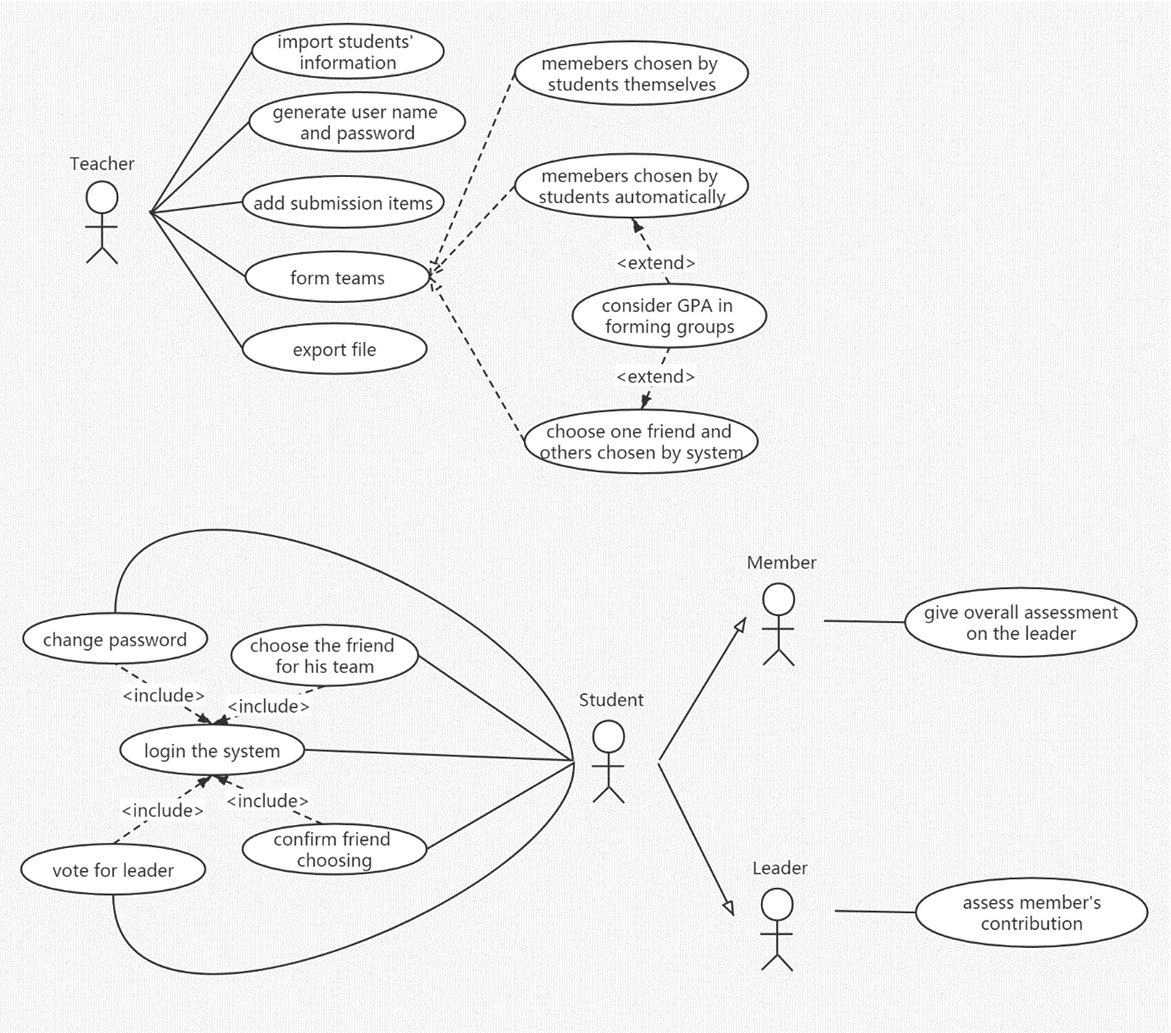
## Product Features

The major features this program contains are the following:

For teachers:

* **Import students’ information**: Teacher may import students’ information (student name, I.D., email, GPA) for a course. The information can also be imported as an excel file.
* **Generate username and password**: generate user name (student’s email) and initial password (same for all) for each student whose name is in the list.
* **Add submission items**: Each submission includes a title (description) and an assessment percentage. The name and percentage for each submission can be modified. The sum of all submission percentages should be 100%.
* **Form teams**: Teacher decides the number of members in a team. Furthermore, this application can offer several choices to form teams:
  1. Students themselves choose all the members.
  2. The members are decided by the system automatically.
  3. A student can choose one friend, and others are given by system randomly
  4. GPA might be considered in 1 and 2. Teacher decides it.
* **Export file**: export a file which lists the contribution for each student in the whole class.

For students:

* **Login the system and change password:** After a student input the correct username and password, the student can log in the system and can change password.
* **Vote for leader:** Vote to choose the project leader of the team.
* **Choose the friend for his team:** Choose his friend for his team if teams are formed using choices 1 and 3 above.
* **Confirm team choosing:** Confirm when his friend chooses him as a friend using choices 1 and 3 above.
* **Assess member’s contribution:** The project leader has the right to assess a member’s contribution in each submission if he likes.
* **Give overall assessment on the leader:** Each member has the right to give the overall assessment on the project leader after the project is finished

User case model of Teamwork System

## User Classes and Characteristics

For the conventional reason, we name each of the user classes-actors with this format:

* **Student:** The student is the one that T.S. to form groups and manage the group, such as the election of team-leaders and making an assessment for each other. It is convenient to think that every student represents every user that contributes to a team individually with an assessment from each other.
* **Teacher:** The Teacher is the one that uses the T.S. to initialize students’ accounts, manage submissions in the courses, forming teams and calculate the final assessment of the students. It is convenient to think that every teacher represents every user that lead the corresponding students and know each one’s contribution separately.

## Operating Environment

This program will operate in the following operating environment for the client and the server GUI:

* Microsoft Windows
* Apple Mac OS X
* Other Linux-based systems

## Design and Implementation Constraints

This program is created using python programming language and uses the Django framework for the main modules. Therefore, a minimum P.C. having at least 64mb of RAM and CPU over 400mhz is required to run the online program with good speed. Or the connection stream TCP-IP is used as its the standard gateway for internet applications.

For natural language, the entire system only supports the English language. The UIC organization has full responsibility for maintaining the Teamwork system.

## User Documentation

Currently, there is no user documentation.

## Assumptions and Dependencies

Django framework was used to create the website application and set up the core program. For better understanding the system, we assume the reader is not interested for how the program is coded. We also assume that we have the right to access the account database in UIC to acquire the user information.

# 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.>

3.1.1 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).>

3.1.2 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.>

3.1.3 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)

# 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

* **User end:**

Smartphones or computers with web connection.

* **Server end:**

A computer server with robust devices (HDD, SSD, etc.) for enough memory space to store data.

## Software Interfaces

|  |  |  |
| --- | --- | --- |
| **Software Type** | **Software** | **Description** |
| Operating system | Windows 10 | We have chosen the Windows operating system for its user-friendliness. |
| Database | MySQL-like database (MariaDB) | To save students and teachers’ records, we recommend using the MariaDB database as it can be set up easily. |
| Libraries and frameworks | Python 3.6.4(32-bit/64-bit), Django (2.0+) | - |
| Webserver | Nginx (1.0+) | - |

## Communications Interfaces

* This project supports all types of web browsers.
* All features will be tested in Google Chrome (73.0.3683.86(64-bit) or later) based on the system of Microsoft Windows 10.
* All traffic should be encrypted by HTTPS. When running locally, do not enable port mapping to ensure data security.
* The ideal data transfer rate for the smooth user experience is 500kbps-1Mbps. The website will use synchronous requests.

# Other Nonfunctional Requirements

## Performance Requirements

Within the scope of server processing capabilities, all requests should react accurately, and the order of execution should be scheduled in the order in which the requests are delivered.

## Safety Requirements

* All user passwords need at least 8 bits with capital or lower-case letters, numbers.
* The transmission of data and the database will use MD5 encryption.
* Save User Information ChangeLog.
* This program uses object-oriented mechanisms to protect its data passed using methods.

## Security Requirements

This program uses object-oriented mechanisms to protect its data passed using methods:

* In term of privacy, all the students could only see their contribution and could not see the contribution of others in the group.
* Students and teachers can log in and change their passwords to keep their accounts safe.

## Software Quality Attributes

* Correctness: All content and information should be correctly displayed and encoded properly. All links need to be correctly connected to the specified location.
* Portability: The website will support new versions of the related browsers.
* Usability: The GUI should be easy to learn and use by the user of any technical background.
* Availability: Checking that the system always has something to function and still pop up error messages in case of component failure. In that case, the error messages appear when something goes wrong so to prevail availability problems

# Other Requirements

Currently, no other requirements.

Appendix A: Glossary

T.S.: Teamwork System

Appendix B: Analysis Models

-

Appendix C: Issues List

-