Software Requirements Specification

for

Teamwork System

Version 1.0 approved

Prepared by Tian Zhanming, Zhang Qiaomu, Zhang Haibin, Li Jiajia

Software Development Workshop III

2020/3/14

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 2

1.5 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 4

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

*3.* System Features 5

3.1 Login 5

3.1.1 Description and Priority 5

3.1.2 Stimulus/Response Sequences 6

3.1.3 Functional Requirements 6

3.2 Accounts import and generation 6

3.2.1 Description and Priority 6

3.2.2 Stimulus/Response Sequences 7

3.2.3 Functional Requirements 7

3.3 Editing submission items 7

3.3.1 Description and Priority 7

3.3.2 Stimulus/Response Sequences 8

3.3.3 Functional Requirements 8

4. External Interface Requirements 9

4.1 User Interfaces 9

4.2 Hardware Interfaces 13

4.3 Software Interfaces 13

4.4 Communications Interfaces 13

5. Other Nonfunctional Requirements 14

5.1 Performance Requirements 14

5.2 Safety Requirements 14

5.3 Security Requirements 14

5.4 Software Quality Attributes 14

6. Other Requirements 14

Appendix A: Glossary 14

Appendix B: Analysis Models 14

Appendix C: Issues List 14

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Zhang Qiaomu, Tian Zhanming,  Zhang Haibin,  Li Jiajia | 2020/03/08 | Prepare initial version | 1.0 |
| Zhang Qiaomu, Tian Zhanming,  Zhang Haibin,  Li Jiajia | 2020/3/14 | Prepare section 3 and section 4.1 | 2.0 |

# 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.), which is developed by Eagle. It also describes non-functional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

## 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.

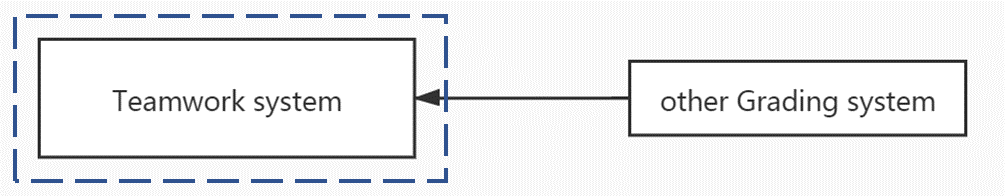
As the description above, the goal and objective of the software is set as an automatically-calculate assessment function. With it, teachers won’t need to do the calculation by themselves, which may even cause some calculation mistakes.

## References

By now, no document is needed to reference.

# Overall Description

## Product Perspective

The teamwork system exports the contribution file will be used in another existing Grading system.

Context model of Teamwork System

## Product Features

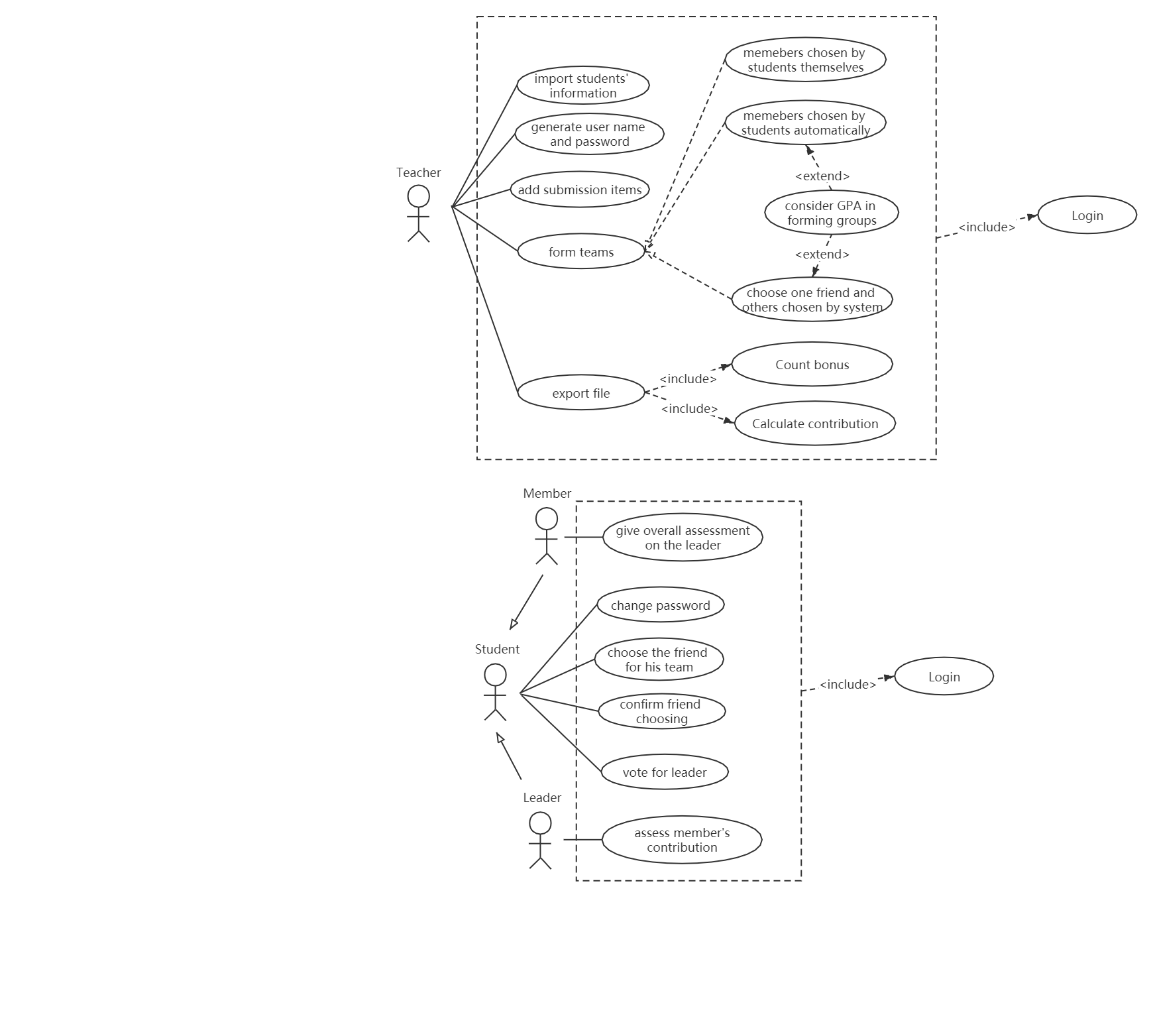
The major features this program contains are the following:

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 in 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

## Login

### Description and Priority

Login has the highest priority. Students and teachers use the same login page to login to their own main page and only if they login can they use other features of this system.

User should first input their username and password. After pressing the “login” button, the system will search the corresponding user in the database. If no user matches or the password is wrong or empty, a warning message shows up. If username and password match the data in the database successfully, users will get into their main page according to their position (student or teacher).

### Stimulus/Response Sequences

### Functional Requirements

REQ-1: After pressing the “Login” button, if the input username and password are empty, warning message “Please input your username and password!” shows up.

REQ-2: After pressing the “Login” button, if the input username doesn’t exist in the database, warning message “No such user! Please input username again.” shows up.

REQ-3: After pressing the “Login” button, if the input password doesn’t match the input username in database, warning message “Password wrong, please input again!” shows up.

REQ-4: After pressing “Login” button, if the user login successfully, as there are two different main pages (teacher’s main page and student’s main page), the corresponding main page will jump out based on the position of the user.

## Accounts import and generation

### Description and Priority

Accounts import and generation have the highest priority. Only if the teacher is login and a course is chosen can let the teacher manage the accounts.

The teacher will have several courses’ link for them to manage those courses further. In a course page, the teacher press “Generate students” to enter the student generate page. Then the teacher can choose to add students into the list by importing an excel file or by inputting the student’s information individually. The input information of the students will be listed out. Teachers should set up an initial password for all the generated students. After confirming the list and initial password, press “Generate” button and finish generation.

### Stimulus/Response Sequences

### Functional Requirements

REQ-1: After pressing “submit” in “Import file page”, The uploaded file should not be empty or a non-excel file and the content should fit the format (name, ID, email, GPA (optional)). If not, warning message “Wrong import!” shows up and the user can choose to cancel uploading the file or upload an excel file again.

REQ-2: After pressing “submit” in “Individual addition page”, the input information should be in the correct format and not empty (GPA part can be empty). If not, warning message “Incorrect student information!” shows up and the user can choose to cancel the adding or input the information again.

REQ-3: After pressing “Generate”, student list should not be empty. If empty, the system will jump to the course page without doing any other things and the warning message “No new students generating” shows up.

REQ-4: Besides an empty student list, if there are students in the list and no initial passwords for generation, a warning message “No initial password!” shows up and the user can choose to cancel the modifying or add an initial password.

## Editing submission items

### Description and Priority

Editing submission items have the highest priority. Only if the teacher is login and a course is chosen can let the teacher manage the accounts.

A teacher can add submission items to the course. After getting into the teacher course page, a teacher can press “modify submission item” to start modifying the whole list of submission items in this course. Inside the submission list modifying a page, the teacher may add submission items or modify the submission items on the existed table. If press “add submission item”, then after input title and percentage, the submission item would be generated. If the teacher wants to modify information in this table, the teacher may directly modify the information, including title and percentage or delete the submission item.

### Stimulus/Response Sequences

### Functional Requirements

REQ-1: After pressing “Confirm” button, if the sum of the percentage is not 100%, the system will show up a warning message “The sum of the percentage of submissions is not 100%” and provide two choices: Go back to the submission list modification page or jump to the course page without changing the submission list that was just modified.

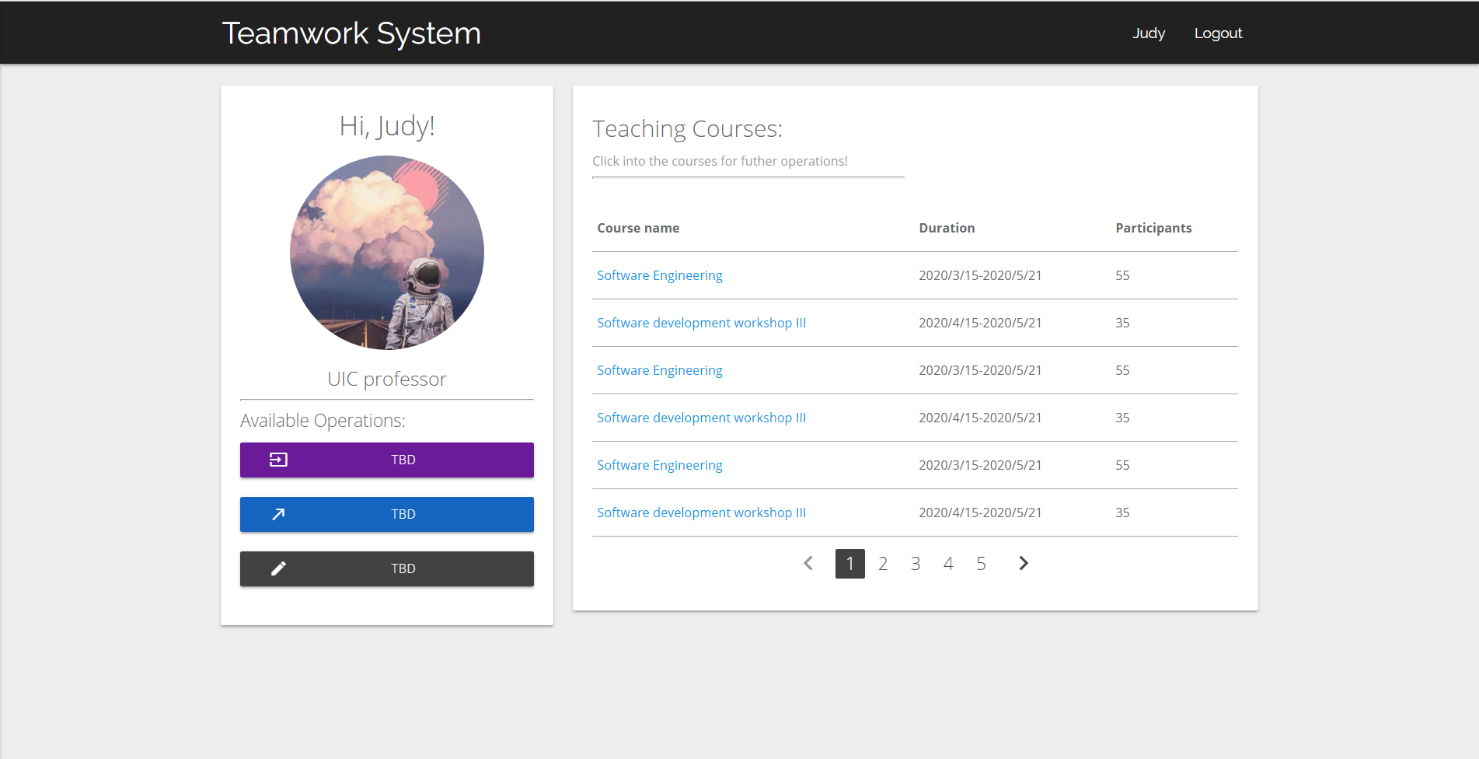
REQ-2: After adding or modifying one submission, if the title is empty, the system will show up a warning message “The title of this submission is empty”. if the percentage is empty, the system will show up a warning message “The percentage of this submission is empty”.

REQ-3: After modifying one submission, if the title is empty, the system will show up a warning message “The title of this submission is empty”.

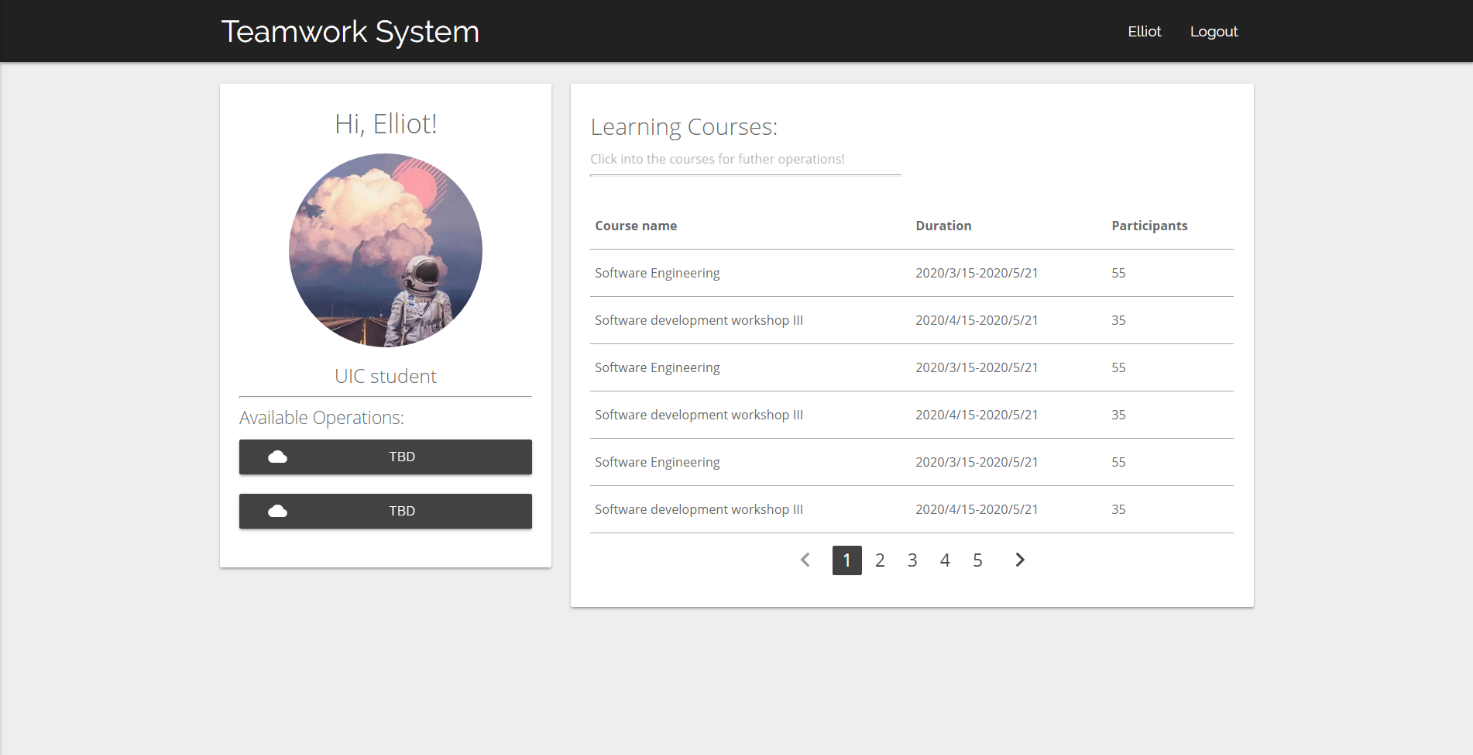
# External Interface Requirements

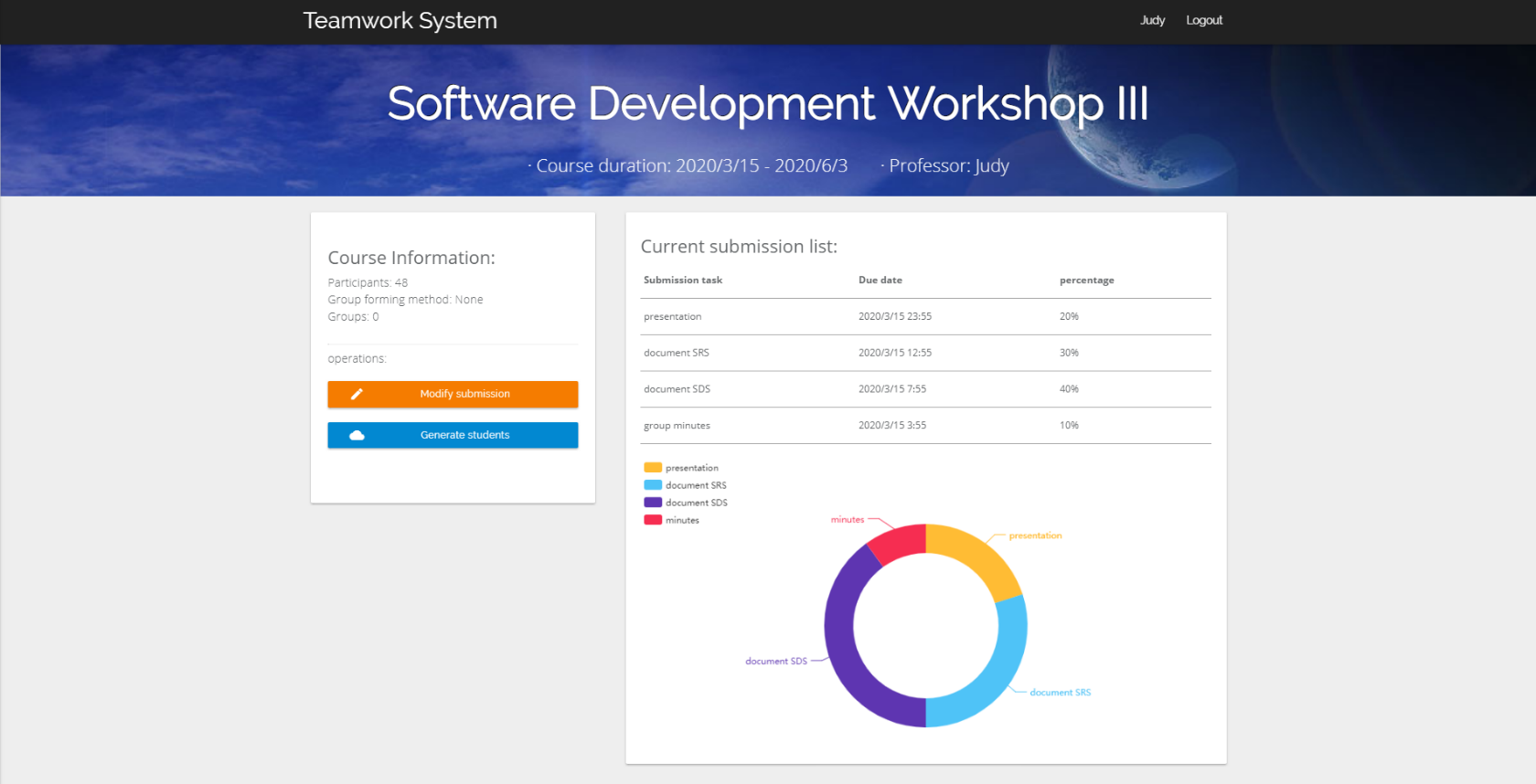
## User Interfaces

1: Login page

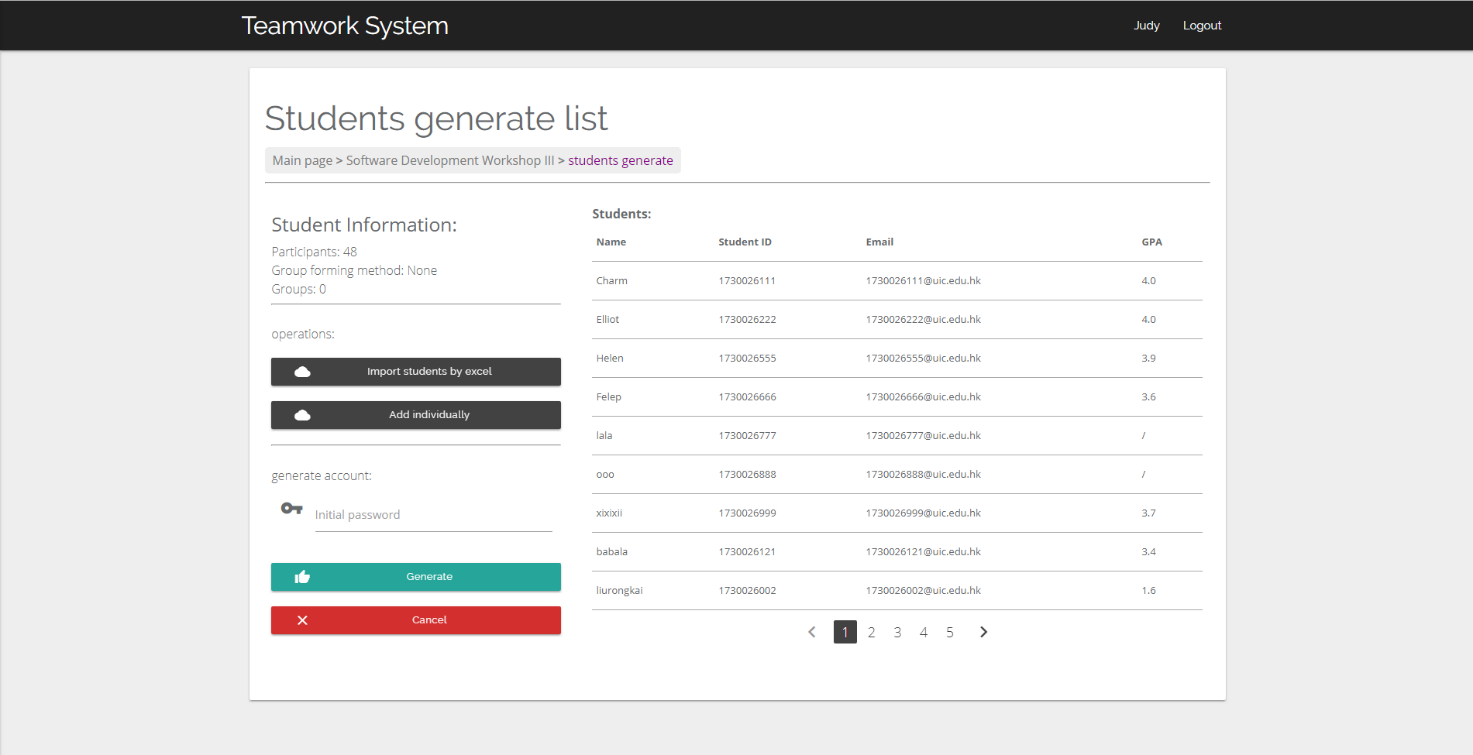


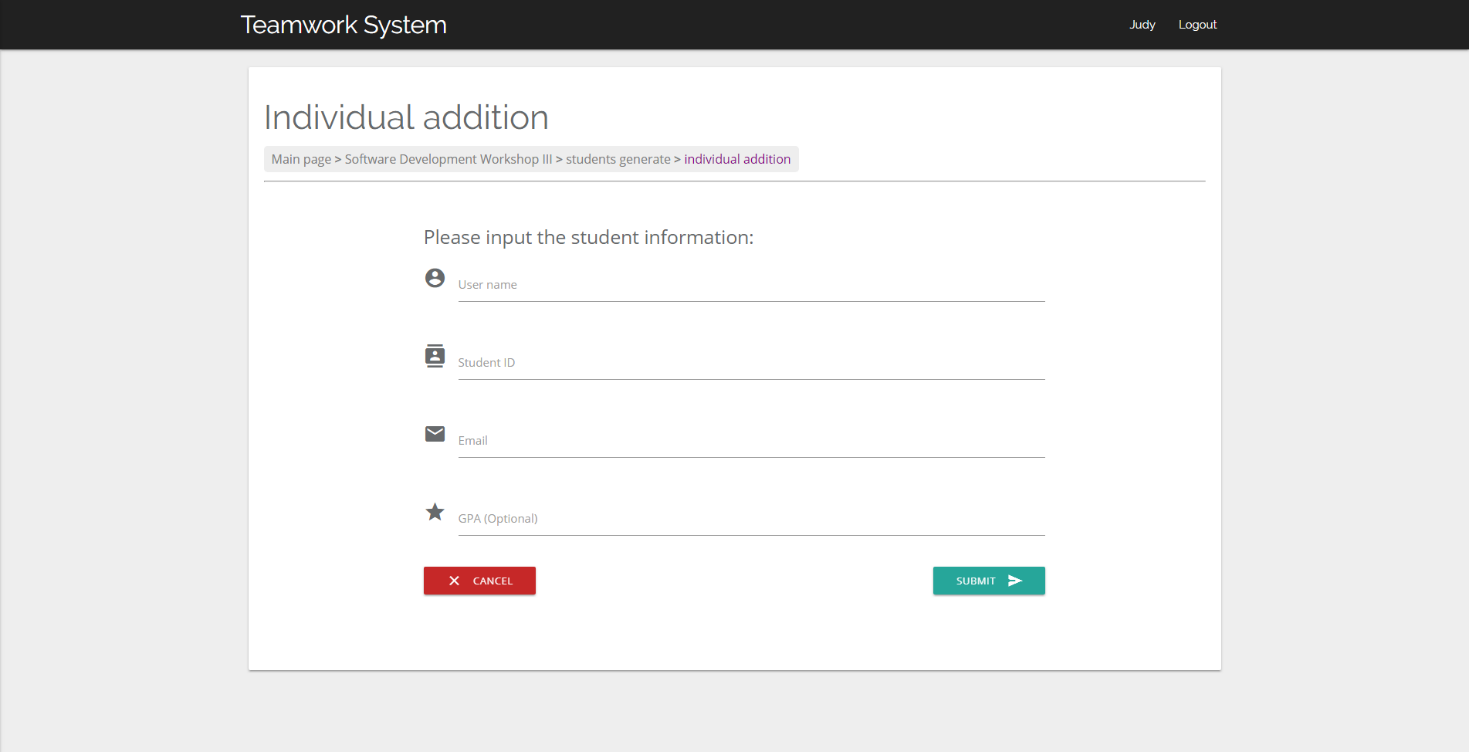
2: Teacher main page

3: Student main page

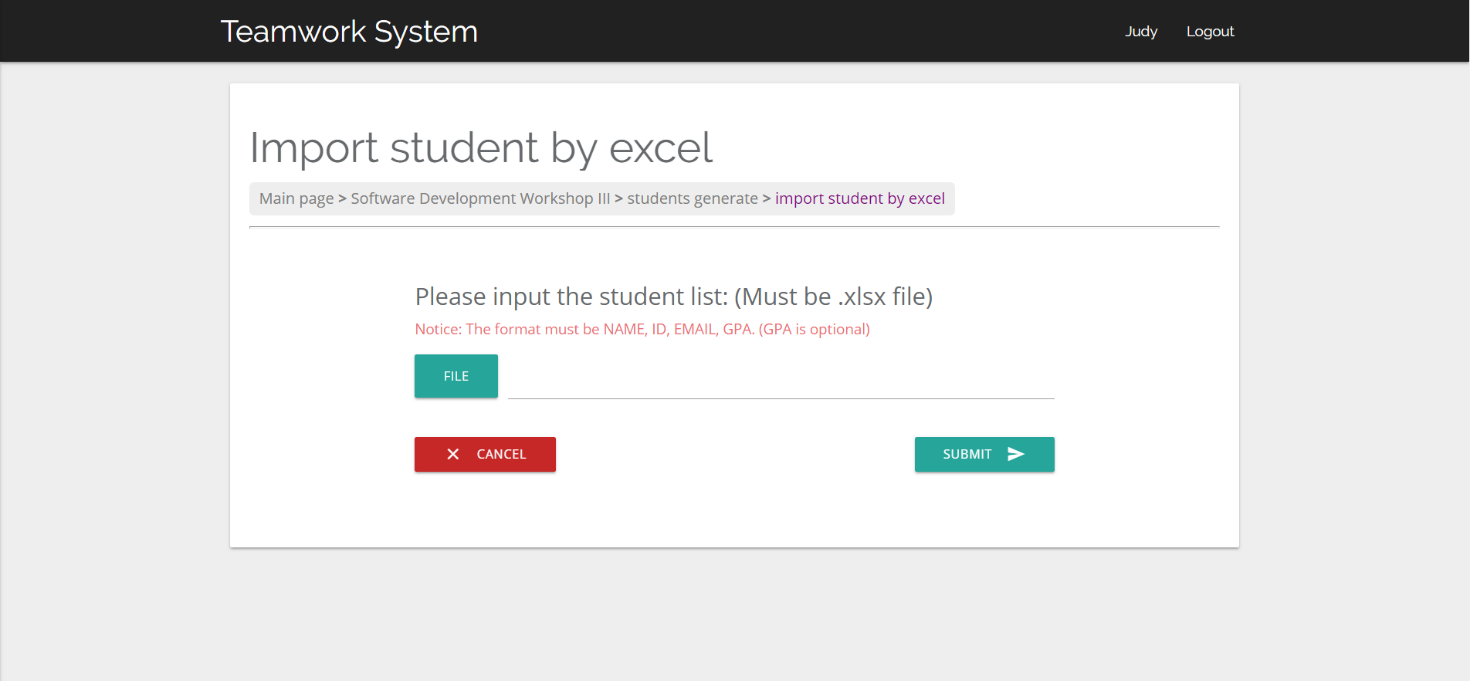


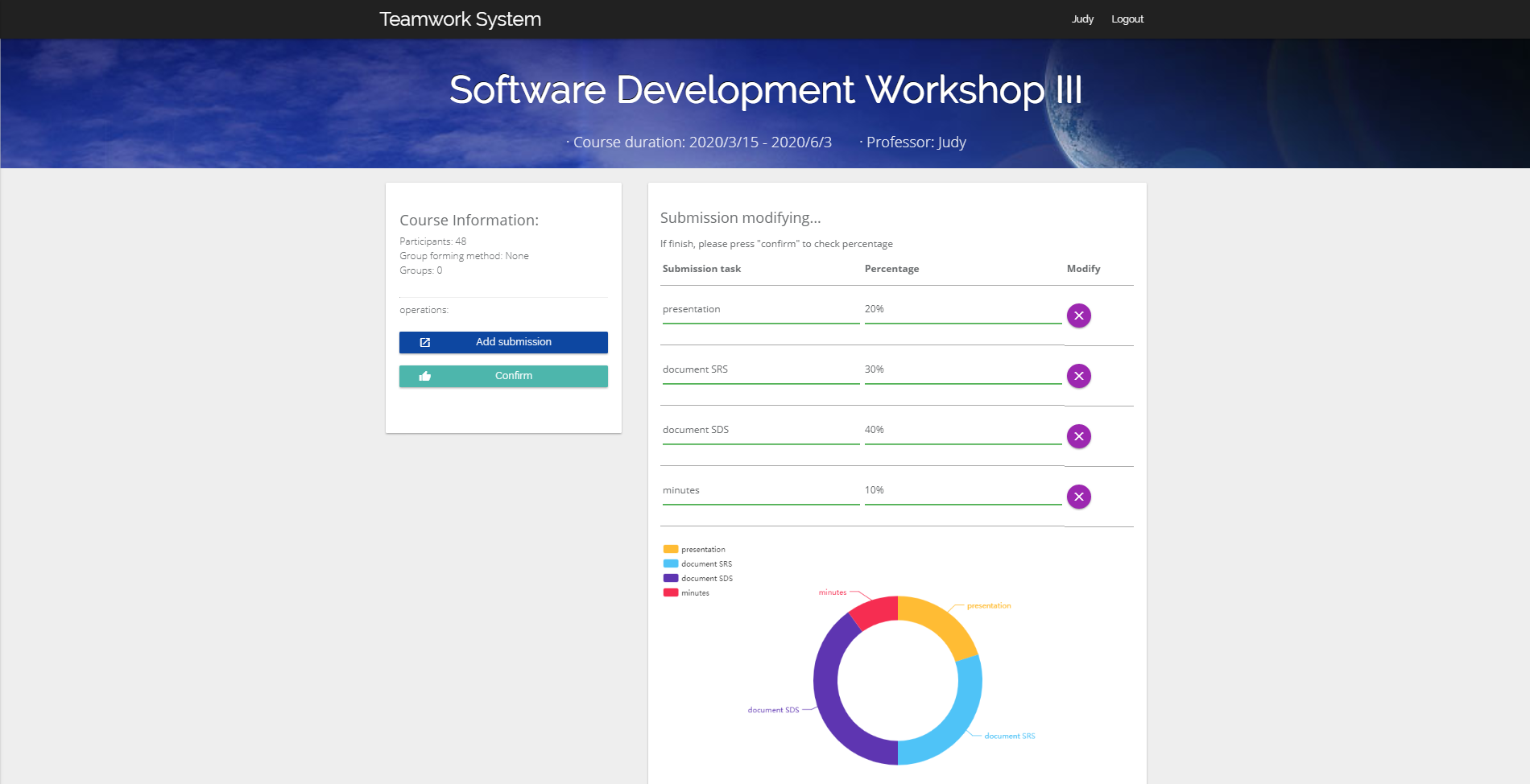
4: Teacher course page

5: Student generate page

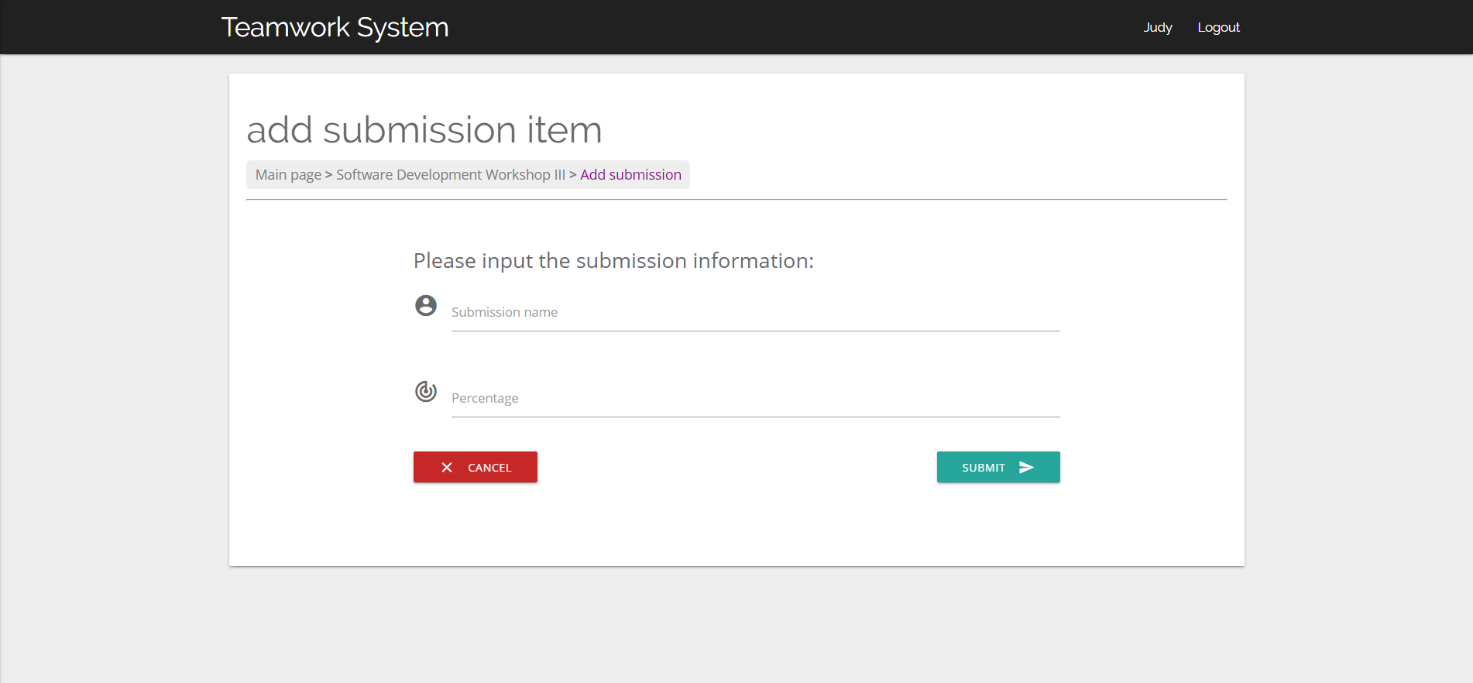


6: Individual addition page

7: Import file page



8. Submission list modifying page

9: Add submission item page

## Hardware Interfaces

None.

## 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

None.

## 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

-