

**Computer Science and Engineering**

**Give&Take**

**System Requirements Specification (SRS)**

# Version 3.0

Document Number: SRS-003

Project Team Number: B13

Project Team Members: Jin Zhou(jz3928), Carrie Song(ys3781), Heng Wu(hw2329), Zhaoyan Zhu(zz2356)

**VERSION 3.0 <11/29/2022>**

# REVIEW AND APPROVALS

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Members** | **Function (Author, Reviewer,**  **Approval)** | **Date** | **Signature** |
| Jin Zhou | Author | 11/29/2022 | JZ |
| Carrie Song | Author | 11/29/2022 | CS |
| Heng Wu | Author | 11/29/2022 | HW |
| Zhaoyan Zhu | Author | 11/29/2022 | ZZ |

**REVISION LEVEL**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
| 10/06/2022 | Version 1.0 | Initial Release |
| 10/26/2022 | Version 2.0 | Requirements Update |
| 11/29/2022 | Version 3.0 | Final Release |

# TABLE OF CONTENTS

1. [DOCUMENT PURPOSE 1](#_DOCUMENT_PURPOSE)
   1. [PURPOSE 1](#_TOC_250037)
2. [INTRODUCTION 2](#_TOC_250036)
   1. [SCOPE 2](#_TOC_250035)
   2. [IDENTIFICATION 3](#_TOC_250034)
   3. [BOUNDS 3](#_TOC_250033)
   4. [OBJECTIVES 4](#_TOC_250032)
   5. [CONTEXT DIAGRAM 5](#_TOC_250031)
   6. [ADDITIONAL DESCRIPTIVE ITEMS 5](#_TOC_250030)
3. [GLOSSARY 7](#_TOC_250029)
4. [REFERENCE DOCUMENTS 7](#_TOC_250028)
5. [BUSINESS REQUIREMENTS 8](#_TOC_250027)
   1. [TECHNOLOGY 8](#_TOC_250026)
   2. [ECONOMICS 8](#_TOC_250025)
   3. [REGULATORY AND LEGAL 8](#_TOC_250024)
   4. [MARKET CONSIDERATIONS 9](#_TOC_250023)
   5. [RISKS AND ALTERNATIVES 9](#_TOC_250022)
   6. [HUMAN RESOURCES AND TRAINING 9](#_TOC_250021)
6. [USER REQUIREMENTS 10](#_USER_REQUIREMENTS_(DESCRIPTIVE)
   1. [FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTS 10](#_TOC_250020)
   2. NON-FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTS 12
7. SYSTEM ARCHITECTURE 13
8. DETAILED SYSTEM REQUIREMENTS – USE CASES 14
   1. REQUIREMENT USE CASES 14
      1. Use Case Diagrams 14
      2. Use Case Descriptions 15
9. SYSTEM MODEL (UML) 22
   1. STATIC - CLASS DIAGRAMS 22
   2. DYNAMIC - BEHAVIORAL MODELS 23
10. [EVOLUTION OF THE SRS 2](#_TOC_250010)4
11. [RATIONALE 2](#_TOC_250009)4
12. [NOTES 2](#_TOC_250008)5
13. [APPENDICES 26](#_APPENDICES)
    1. [SYSTEM TEST PLAN REQUIREMENTS 26](#_TOC_250006)
    2. [QUALIFICATION PROVISIONS 27](#_TOC_250005)
    3. [REQUIREMENTS TRACEABILITY 27](#_TOC_250004)
    4. [SCHEDULE TRACKING 28](#_TOC_250003)
    5. [DEFECT TRACKING 30](#_TOC_250002)
    6. [DICTIONARIES 32](#_TOC_250001)
14. [INDEX………………………………………………………………………………………………………………………………………………..32](#_TOC_250000)

# DOCUMENT PURPOSE

# Purpose

The purpose of this SRS document is to outline the requirements and different aspects of the Give&Take Web Application, as well as all the necessary steps to develop this application.

The first part of this document is to introduce the project which covers the Scope, Identification, Bounds, Objectives, and Context Diagram. In this part, the audience (usually engineers and stakeholders in the real industry) will have a glance at the software architecture of this project being implemented. The next few parts will cover business and user requirements. The business requirements section includes aspects from Technology and Economics impact to Regulatory/Legal components and Market Considerations, to Risks and Human resources training. The User Requirements part contains all the software requirements to a level of detail sufficient for the designers, engineers, and testers to be able to test all the requirements. This section will be gradually updated with new requirements and progresses.

The audience for this SRS document would be people who need to implement/test the Give&Take Web application, and people who are interested in understanding this project in detail.

# INTRODUCTION

# Scope

The Give&Take project is intended to focus on global environmental issues, specifically taking care of the waste of worn clothes. The project shall build an online platform where people can donate unused clothes and request clothes based on their needs. The project shall store users’ personal information securely in the database. The project shall construct a non-commercial consumption system, ensuring donors can receive rewards based on their contribution and requesters can purchase clothes successfully without using actual money. The project shall build a communication system, enabling users to ask and answer questions and leave rating and reviews. The project shall establish an activity profile for users where they can view their past activities like previous transactions. In summary, the project shall include the following functionalities.

* + 1. Registration of New Users
    2. Users’ Profile
    3. Authentication of Suppliers and Requesters
    4. Browsing and Jump to Different Pages
    5. Clothes Donation and Clothes Condition Check
    6. Clothes Request
    7. Non-Commercial Transaction
    8. Interaction between Users
    9. Ranking and Yearly Reports
    10. Other Website-Based Activities

The Give&Take project will not allow users to make transactions using methods other than website credits. The project will not promote commercial personalized advertisements to users. The project will not permit users to make transactions without the surveillance of the website.

# Identification

*Document Name*: Give&Take System Requirement Specification

*Document Number* : SRS-003

*Revision Number* : Version 3.0

# Bounds

The system aims to be available globally. The first version of the project should be to be tested within the US with coordination of a few large charity groups with the main function being donation of clothing. The project is a clothing-focused system that targets to help problems with clothe wasting and people in need of clothes.

# Objectives

The project’s priority is to establish the function to donate clothes to charity groups and registration for accounts to do so. The communication with charity groups should also be prioritized to set up requirements for requester accounts which can then initialize donation requests on the system.

The project will be using an incremental life cycle approach. With the development approach, the system will be able to adapt with spontaneous changes due to factors such as feedbacks on the system, suggestions from charity groups, etc.

*Initial deliverables*:

Project proposal 09/22/2022

Software Business Specification (SRS – Business Definition) 10/06/2022 Software Requirement and Analysis Specification (SRS - Requirements) 10/27/2022 Software Project Management Plan (SPMP) 11/10/2022

Project Presentation 12/01/2022

Software Analysis Specification – Final (SRS – Analysis) 12/01/2022

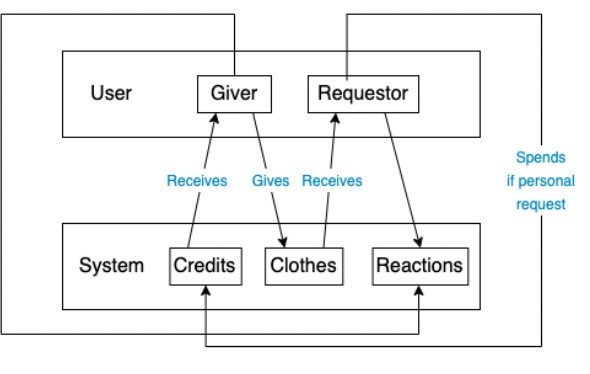
Project Description 02/22/2023

Software Design Document (SDD) 03/01/2023

Implementation (code with documentation) 05/01/2023

Presentation and Demonstration 05/01/2022

# Context Diagram



# Additional Descriptive Items

### Product functions

A system that can enable people to purchase used clothes from others with the system credits and donate their own used clothes to some charity organizations or list their used clothes for others to buy.

All kinds of users can view different categories of items, and they can see all ongoing requests and participate in them.

Each user can be an individual requester and supplier.

Individual requesters can purchase clothes by spending their credits. Organization requesters can initiate a donation request for free.

Suppliers can list their clothes for others to purchase.

### User characteristics

This system is designed for people who have some used clothes but they don’t want to just throw them away, and they hope to find someone who wants their used clothes or donate to the charity to help others. The users need to have some basic knowledge about computers. They should have some familiarity with the web. They also need to have some basic experience about how to use an online shopping system.

### Constraints

*Time*:

The project needs to be completed before May 2023, some additional features may not be implemented because of the time constraint.

*Hardware limitations and budget*:

The local computers may not have enough space to hold a lot of user information.

We may need some extra budgets for the cloud system to store user data.

*Development languages*:

Some developers in the team may not be familiar with the language such as SQL or HTML.

*Safety and security*:

The user information such as password may not be protected well in this system because of the limited budget and time to implement a highly secured system.

### Assumptions and dependencies

The application is designed for the PC users.

No user will transfer clothes between his/her own multiple accounts in order to earn more credits.

Each user has a good understanding of how to create a secured password.

The system will never receive any suspicious attack to steal or change users’ data.

### Requirements subsets

Duo authentication during login. Check for suspicious transactions.

Reward for users who have a large number of credits.

# GLOSSARY

### Credits:

The currency system in this project. The requesters use credits to purchase clothes, and the suppliers can earn credits for each clothes they sold.

# REFERENCE DOCUMENTS

A13, Project Proposal, 9/22/2022

A13, System Requirements Specification, SRS, Version 1.0, 10/06/2022

A13, System Requirements Specification, SRS, Version 2.0, 10/26/2022

A13, Software Project Management Plan, SPMP, Version 1.0, 11/07/2022

A13, System Requirements Specification, SRS, Version 3.0, 11/29/2022

# BUSINESS REQUIREMENTS

# Technology

To many people, they have a mindset that worn clothes should be discarded; in other words, few of them have the idea to donate these clothes to others who need them. Even though some people may realize the huge adverse impact brought by clothes decomposition on the environment and they decide to donate clothes, they do not know how to achieve that. Even if they know how to donate clothes, they may consider the process as annoying and so they choose not to. Moreover, many poor people who cannot afford clothes may need these worn clothes. In order to provide people a convenient way to donate clothes and the poor an accessible way to get clothes, we decided to build this website. We hope this website will make a positive impact at any level.

# Economics

The Give&Take project has one main economic impact: supporting people who are suffering clothing insecurity. Every year, especially during winter, some people have no money to buy clothes like down-filled clothes. The project can save their money and provide them with needed clothes. Clothing is always a basic necessity of living. It is more likely to achieve economic development if people’s basic needs are satisfied.

# Regulatory and Legal

For this system, there is currently no regulatory and legal requirement.

# Market Considerations

The current market of this system is for the individual PC users in the US. To expand the market, this system needs to be customized for users in other countries. Based on the different situations and user habitats of different countries, we can modify the current system to fulfill different needs.

The mobile versions such IOS or Android of this system can also be implemented in the future for more convenient use.

This system can also be used by the government for requesting large amounts of clothes to support poor families or the homeless. Government may also use our system to reduce fabric waste.

# Risks and Alternatives

### Financial Risk:

Since we will be responsible for the delivery of the clothes, and the project is non- profit, we will have to empty our own pockets to support the project.

Solution: Requesting donations from the charities and users, additionally, offer advertisement opportunities in return for funding

# Human Resources and Training

At this point, there will be no major training but software development and project management. All the human resources are from the product development team which consists of four people. However, there will be more human resources such as volunteers who can help verify and classify all the old clothes needed in the future.

# USER REQUIREMENTS (DESCRIPTIVE FUNCTIONAL AND NON- FUNCTIONAL REQUIREMENTS)

# Functional Descriptive Detailed Requirements

**Donation System:**

Every user can enter the donation section. Donators can enter the donation section and upload pictures of their items, as well as the description of conditions. The system will detect then entrance (donation), grab all the data (item information and users’ personal information) and save them to the database. The system will ask the user to choose a category for each item before uploading to help the filtering process.

**Receiver System:**

The receiver system consists with two different parts. One is for individual users, and the other is for large organizations / charities. The system will require large organizations to go through a more complicated process since they can have more benefits than normal users do. The receiver system allows users to search for the item they are looking for.

**Filtering system:**

The filtering system allows user to filter the items by different categories such as clothing or educational materials. Users will be able to select their desired items by using this feature.

**Trading System:**

The trading system works similarly to those of shopping websites. The system allows receivers to “buy” the items they need, and the donator will “sell” them. Every time when one trade is completed, the number of the user’s trading frequency will increment correspondingly. The system will grab the trading information and save the frequency to the user’s database. Additionally, every user will need to provide their personal contact information to complete a trade.

**Credit System:**

The credit system will cumulate the number of successful trades of each user according to the trading system & the database. The credit system will automatically rank the loyal users and the end of each season / year, it will push a notification to the user notifying them of their ranks and the benefits they get.

**Communication System:**

The communication system will allow users to chat when a trade is ongoing. The receiver can ask the donator to provide more detailed information about the item and the donator also have the right to reject. Additionally, the system will also allow official notifications to pop up.

# Non-Functional Descriptive Detailed Requirements

# Product requirements:

The credit feature will require continuous tracking of the number of transactions of each user and will need to update spontaneously. The trade feature will also need to keep updating the status of the item. If the stock gets to 0, the status will need to be updated as unavailable, and the purchasing button should be disabled. Delay in updates during peak hours should not exceed more than 1 minute. Delay in updates during non-peak hours should not exceed more than 5 minutes. More data and research are required to identify the peak and non-peak hours. Since users are free to upload the items they want, the uploaded item will need a check before it’s visible to all the users to prevent dangerous items.

**Organization requirements:**

Users who want to donate / receive items will need to register an account with valid personal contact information including name, phone number, home address, and email address. Large organizations will need to provide official documents to prove their authenticity.

**External requirements:**

The trading feature will secure user data. Sensitive information such as phone number and email address will be kept confidential. Only when the users completed a trade successfully, they can view each other’s personal information.

# SYSTEM ARCHITECTURE

Graphical user interface, application, Teams

Description automatically generated

In the diagram above, the users of the website are divided into three groups: website administrators, clothes suppliers, and clothes donators. Each group can access to different interfaces: for website administrators, they can access the administrator’s interface; for other two groups, they can access the user’s interface. All these interfaces are connected to the website server, and so different users can use different functionalities the website provides. There is also a database which saves all data, such as the login information and personal data, from the website. Also, the database saves all edits and updates and can delete data based on users’ operations.

# DETAILED SYSTEM REQUIREMENTS – USE CASES

# Requirement Use Cases

## Use Case Diagrams

Diagram

Description automatically generated

## Use Case Descriptions

|  |  |  |
| --- | --- | --- |
| **Use Case**  **Browsing** | | |
| **Description** | All users shall be able to explore the website with or without login. They shall be able to search for items, checking item descriptions and view user ratings. | |
| **Pre- Conditions** | Users shall have a device that connected to internet and a browser that can work successfully. | |
| **Flows** | **Basic or** | 1. Connect the device to internet. |
|  | **Normal Flows** | 2. Enter the website address.  3. Explore the website. |
|  |  |  |
|  | **Alternative** | 1. When a user disconnected from the internet. It is ejected to the customer with “Internet disconnected” message. The use case ends. |
| **Flows** | 2. When a user attempts to view information that can only be shown to logged users, they will be directed to the login page. |
|  |  |
| **Post Conditions** | At the end of the use case, if the user doesn’t login, no information on any page shall be modified. | |
| **Special Requirements** | The page loading speed shall be quick, no more than 3 seconds with a good internet connection. The website shall not use a lot of internet resources that may slow down users’ internet speed. | |
| **Extension Points** | No extension points currently. | |

|  |  |  |
| --- | --- | --- |
| **Use Case**  **User register** | | |
| **Description** | Users shall be able to register their own unique account by entering necessary information. | |
| **Pre- Conditions** | Users have a valid email address and phone number.  Users have an internet connection. | |
| **Flows** | **Basic or** | 1. Connect the internet. |
|  | **Normal Flows** | 2. Enter the website address.  3. Click “Register”.  4. Enter the email address.  5. Enter the username they want to use.  6. Enter and re-enter the password.  8. Click “Finish Register” |
|  |  |  |
|  | **Alternative** | 1. If the user’s email address has already been used. Resume at step 4, with a pop-up message “The email has already been registered”. |
| **Flows** | 2. If the user re-entered a password that doesn’t match the first one, resume at step 6 with a pop-up message “password doesn’t match”. |
|  | 3. If the user clicked “Finish Register” with some information empty, resume at those steps pop-up message “Information needed”. |
| **Post Conditions** | The user shall receive a confirmation after register. When the user confirmed, their register information shall store into the database, and they can login successfully. | |
| **Special Requirements** | The Register page shall be clear, simple and easy to understand with some necessary aesthetics.  The system shall report the error accurately to the user.  The confirmation shall be sent to the user within 1 minute. | |
| **Extension Points** | The user should be able to register with their mobile phone number or linked the phone number to their account. Therefore, they shall receive a confirmation text message in their phone. | |

|  |  |  |
| --- | --- | --- |
| **Use Case**  **Login** | | |
| **Description** | The user can log into their account after successfully registered. | |
| **Pre- Conditions** | The user has a good internet connection.  The user has a successfully registered account. | |
| **Flows** | **Basic or** | 1. Click “Login”.  2. Enter the email address.  3. Enter the password.  4. Click “Login” on bottom of the page. |
|  | **Normal Flows** |  |
|  |  |  |
|  | **Alternative** | 1. The user doesn’t have a valid account. The page should eject a pop-up message “The account doesn’t exist” with an option to direct the user to the register page.  2. If the user entered the invalid password, resume at step 3. |
| **Flows** |  |
|  |  |
| **Post Conditions** | The account information should be authenticated through the database. After login, the user shall be able to see their account information. | |
| **Special Requirements** | The maximum allowance of entering invalid passwords is 6 times. After the user entered the incorrect password more than 6 times, their account will be locked for 12 hours, and they can’t login in during the locking phase. | |
| **Extension Points** | The user should be able to select login method. The user should be able to choose login with their phone number, and they will receive a message on their phone containing the dynamic code, they can use the dynamic code to login. | |

|  |  |  |
| --- | --- | --- |
| **Use Case**  **List item** | | |
| **Description** | The user shall be able to list the item after login and set a price based on credits. They shall include the item description and photos. | |
| **Pre- Conditions** | The user has a good internet connection.  The user has a valid account. | |
| **Flows** | **Basic or** | 1. Click “List item”.  2. Select the type of the item, such as “Jacket” or “T-shirt”.  3. Select the size of the item.  4. Upload photos.  5. Upload additional descriptions. |
|  | **Normal Flows** | 6. Set up the price.  7. Click “Finish listing”.  8. Receive the confirmation email. |
|  |  |  |
|  | **Alternative** | 1. If the user ignores uploading some necessary item information when click “Finish listing”, they will receive a pop-up message to remind them add those information and resume at those steps. |
| **Flows** |  |
|  |  |
| **Post Conditions** | After listing successfully, the item shall be stored into the database with a unique database, item information and user id who listed it. All users shall be able to see the item by searching. The photos should be clear and no distortion. The price should be same as the user set. | |
| **Special Requirements** | The photo should be able to upload the photos from their own local device or from the cloud.  After finishing listing, they item shall be shown to all users within 5 minutes. | |
| **Extension Points** | The user should be able to relist the item from their item history, and they just need to click “Relist”, the item will be relisted with saved information. | |

|  |  |  |
| --- | --- | --- |
| **Use Case**  **Request** | | |
| **Description** | The user shall be able to purchase the item with their credits. Both the requester and supplier will have accurate balance after the successful purchase. | |
| **Pre- Conditions** | Users have a good internet connection.  Users have a valid account. | |
| **Flows** | **Basic or** | 1. Click the item.  2. Click the “Purchase” button.  3. Edit and select their shipping address.  4. Click “Submit order”.  5. Receive the confirmation email. |
|  | **Normal Flows** |  |
|  |  |  |
|  | **Alternative** | 1. When the requester doesn’t have enough credit balance to purchase this item, they will receive a message “Sorry, you don’t have enough credits for this item”. The use case ends. |
| **Flows** |  |
|  |  |
| **Post Conditions** | After successful purchase, the listed item shall be removed from the market. Only visible to the requester and supplier in their transaction history.  The correct balance of the requester and supplier shall be calculated and store into the database accurately. Both the requester and supplier shall be able to see the latest amount of balance in their account. | |
| **Special Requirements** | If the listed item is no longer available after the requester submit the offer, the supplier shall notify the requester. Both the requester and supplier shall be able to cancel the order, and their balance shall be recalculated correctly and store into the database. | |
| **Extension Points** | The requester can choose whether to “make an offer” on the item page. They can send their offered price to the supplier, and the system shall send an email to the user about the offer. If the supplier accepts the offer, the requester must purchase this item. | |

|  |  |  |
| --- | --- | --- |
| **Use Case**  **Rate the order** | | |
| **Description** | After successfully purchase an item, the user shall be able to give ratings to this item and the supplier. The rating is visible to all users. | |
| **Pre- Conditions** | The user has a good internet connection.  The user has a valid account.  The user has finished the purchase and received the item. | |
| **Flows** | **Basic or** | 1. Login the account.  2. Click “Order history”.  3. Select the received item to rate.  4. Give the general rating (points from 0.0 to 5.0).  5. Add additional description.  6. Submit rating. |
|  | **Normal Flows** |  |
|  |  |  |
|  | **Alternative** | 1. If the internet is disconnected during the rating, users will receive the message “No internet connection”. The use case ends. |
| **Flows** |  |
|  |  |
| **Post Conditions** | The rating is stored into the database, and it is visible to all users. The cumulative rating ratings of the supplier will be calculated, stored into the database and visible to all users. | |
| **Special Requirements** | If the requester doesn’t submit the rating after 30 days of the purchase. The system shall automatically give the 5.0 rating to the supplier. | |
| **Extension Points** | On the rating page, the user has an option of “Report this supplier” to report the illegitimate action of this supplier. The requester shall provide additional evidence to the system for validation. If it is confirmed by the system, the supplier will be banned from the system. | |

|  |  |  |
| --- | --- | --- |
| **Use Case**  **View report** | | |
| **Description** | After login, the user shall be able to see their monthly or yearly report about how many items have they sold, how many credits have they earned and their current credit balance. The user shall also be able to view their rankings based on the total credits they earned. | |
| **Pre- Conditions** | The user has an internet connection.  The user has a valid account. | |
| **Flows** | **Basic or** | 1. Login the account.  2. Select view progress report.  3. Select the type of report to view.  4. Submit the request.  5. Generate the report.  6. View the report. |
|  | **Normal Flows** |  |
|  |  |  |
|  | **Alternative** | 1. If the user wants to change the type of report they want to view, they shall be able to resume at step 2 by clicking “Change report type”. |
| **Flows** |  |
|  |  |
| **Post Conditions** | The database shall select the correct information to retrieve and generate to the user. | |
| **Special Requirements** | The report shall be generated within 30 seconds.  After every new successful transaction, the report shall include the new changes within 24 hours. | |
| **Extension Points** | No extension points currently. | |

# SYSTEM MODEL (UML)

# Static - Class Diagrams

Diagram

Description automatically generated

# Dynamic - Behavioral Models

Event diagrams (one per each Use Case/Function Requirement)

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with medium confidence

# EVOLUTION OF THE SRS

The SRS version 1.0 is first released on October 6, 2022;

The SRS version 2.0 is released on October 26, 2022;

The SRS version 3.0 is released on November 29, 2022.

# RATIONALE

Currently, many people still do not have the habits of recycling, and so they tend to throw away their unwanted clothes. This kind of action causes harm to the environment, because the landfilling and burning process of clothes release toxic substance and also waste resources. However, there are still many poor people who cannot afford clothing. Even though some people want to donate their unwanted clothes, they do not have a suitable platform to accomplish this action. Many existing online platforms have really complicated users interface, confused collecting and shipping policies, and unattractive donation rewards, so many people choose not to use these platforms and then throw away their unwanted clothes. Therefore, we hope to construct an online platform which is simple, understandable, and attractive for clothes donators and requesters to use. We hope that clothes donators will no longer think clothes donation is a complex thing after using our website. Also, we hope that clothes requesters can pick and get the clothes they want conveniently from the website. As a result, we wish that all unwanted clothes can be reused; and we really hope that our project can make a positive contribution to the environment protection and economic development.

# NOTES

# APPENDICES

# System Test Plan Requirements

The project will be developed using an incremental approach. The project will go through different validation stages based on its status.

The first phase is the planning phase. Our team plans and establishes the detailed process to achieve the requirements of our project. We will complete specifications, proposals, SRS, and SPMP in this phase.

The second phase is the implementation and validation phase. This phase involves the development and validation of the processes. We need to develop all functionalities of the system and test them using unit tests and integration tests. For unit tests, these tests test basic functionality, such as regular functions, helper functions, and modification of databases. For integration tests, these tests test the simulation of the website: for example, these tests simulate the interactions between users and the websites. These tests help to ensure whether all functionalities produce desirable outputs or not.

The third phase is the check phase. In this phase, we further check whether the system is operating in the expected way or not. Also, we check whether the website performs correctly based on the users’ actions or not.

The final phase is the fail-safe phase. In this phase, we implement actions which are necessary to achieve improvements in the processes, so we can handle bugs and unexpected break-down easily if these situations happen.

# Qualification Provisions

The four review types include self or deck check, peer review, walkthrough and inspection.

Every team member needs to read their work carefully and to minimize their grammar and logic errors. Also, each team member should check others’ work for correctness and consistency. For walk through review, each member can invite others outside the team to review the documentation and find defects.

# Requirements Traceability

Requirement 2.1.1 (Registration of New Users), Requirement 2.1.2 (Users’ Profile), Requirement 2.1.4 (Browsing and Jump to Different Pages), Requirement 2.1.8 (Interaction between Users), and Requirement 2.1.9 (Ranking and Yearly Reports) can be traced forward.

Requirement 2.1.3 (Authentication of Suppliers and Requesters), Requirement 2.1.5 (Clothes Donation and Clothes Condition Check), Requirement 2.1.6 (Clothes Request), and Requirement 2.1.6 (Non-Commercial Transaction) can be traced backward.

The traceability of Requirement 2.1.10 (Other Website-Based Activities) is based on the specific requirement.

We hope to ensure all requirements can be traced bidirectional.

Version 1.0: All sections except 6-12 and appendices.

Version 2.0: Sections 6, 7, 8.

Version 3.0: Sections 9, 10, 11, 12.

# Schedule Tracking

**Hours**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SRS-Domain | Heng Wu | 2.0 | 1.5 | -0.5 |
| SRS - Domain | Zhaoyan Zhu | 2.0 | 3.0 | +1.0 |
| SRS - Domain | Jin Zhou | 1.5 | 2.5 | +1.0 |
| SRS - Domain | Carrie Song | 1.5 | 2.0 | +0.5 |
| SRS - Domain | Team Total | 7.0 | 9.0 | +2.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SRS-Requirements | Heng Wu | 2.0 | 2.5 | +0.5 |
| SRS-Requirements | Zhaoyan Zhu | 3.0 | 2.0 | -1.0 |
| SRS-Requirements | Jin Zhou | 1.5 | 3.0 | +1.5 |
| SRS-Requirements | Carrie Song | 4.0 | 2.0 | -2.0 |
| SRS-Requirements | Team Total | 10.5 | 9.5 | -1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SPMP | Heng Wu | 2.0 | 1.5 | -0.5 |
| SPMP | Zhaoyan Zhu | 3.0 | 4.0 | +1.0 |
| SPMP | Jin Zhou | 2.5 | 2.0 | -0.5 |
| SPMP | Carrie Song | 3.0 | 2.5 | -0.5 |
| SPMP | Team Total | 10.5 | 10.0 | -0.5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SRS-Analysis | Heng Wu | 3.0 | 1.5 | -1.5 |
| SRS-Analysis | Zhaoyan Zhu | 3.0 | 3.0 | 0 |
| SRS-Analysis | Jin Zhou | 1.5 | 2.0 | +0.5 |
| SRS-Analysis | Carrie Song | 4.0 | 3.0 | -1.0 |
| SRS-Analysis | Team Total | 11.5 | 9.5 | -2.0 |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or Team) | Estimated | Actual | Difference |
| Heng Wu | 9.0 | 7.0 | -2.0 |
| Zhaoyan Zhu | 11.0 | 12.0 | -1.0 |
| Jin Zhou | 7.0 | 9.5 | +1.5 |
| Carrie Song | 12.5 | 9.5 | -3.0 |
| Team Total | 39.5 | 38.0 | -1.5 |

# Defect Tracking

**Counts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SRS - Domain | Heng Wu | 5 | 3 | -2 |
| SRS - Domain | Zhaoyan Zhu | 0 | 2 | +2 |
| SRS - Domain | Jin Zhou | 3 | 2 | -1 |
| SRS- Domain | Carrie Song | 2 | 3 | +1 |
| SRS - Domain | Summary for entire team | 10 | 10 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SRS-Requirements | Heng Wu | 4 | 4 | 0 |
| SRS-Requirements | Zhaoyan Zhu | 1 | 2 | +1 |
| SRS-Requirements | Jin Zhou | 2 | 3 | +1 |
| SRS-Requirements | Carrie Song | 4 | 3 | -1 |
| SRS-Requirements | Team Total | 11 | 12 | +1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SPMP | Heng Wu | 5 | 4 | -1 |
| SPMP | Zhaoyan Zhu | 3 | 2 | -1 |
| SPMP | Jin Zhou | 2 | 4 | +2 |
| SPMP | Carrie Song | 1 | 3 | +2 |
| SPMP | Team Total | 11 | 13 | +2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SRS-Analysis | Heng Wu | 3 | 1 | -2 |
| SRS-Analysis | Zhaoyan Zhu | 4 | 2 | -2 |
| SRS-Analysis | Jin Zhou | 1 | 3 | +2 |
| SRS-Analysis | Carrie Song | 2 | 2 | 0 |
| SRS-Analysis | Team Total | 10 | 8 | -2 |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or Team) | Estimated | Actual | Difference |
| Heng Wu | 17 | 12 | -5 |
| Zhaoyan Zhu | 8 | 8 | 0 |
| Jin Zhou | 8 | 12 | +4 |
| Carrie Song | 9 | 11 | +2 |
| Team Total | 42 | 43 | +1 |

# Dictionaries

*Data Structures / Classes*:

B+ Tree or B- Tree (Large number of children per node); Stack / Queue(Collection of Elements);

Hash Table (Maps keys to values); etc.

# INDEX