

**Computer Science and Engineering**

**Give&Take**

**Software Design Description (SDD)**

## Version 1.0

Document Number: SDD-001

Team Number: B13

Team Members (Name and NET\_ID): Zhaoyan Zhu(zz2356), Heng Wu(hw2329), Jin Zhou(jz3928), Carrie Song(ys3781)

**VERSION 1 . 0 March 2, 2023**

# REVIEW AND APPROVALS

|  |  |  |  |
| --- | --- | --- | --- |
| **Printed Name and Title** | **Function (Author, Reviewer,**  **Approval)** | **Date** | **Signature** |
| Zhaoyan Zhu | Author | March 2, 2023 | ZZ |
| Heng Wu | Author | March 2, 2023 | HW |
| Jin Zhou | Author | March 2, 2023 | JZ |
| Carrie Song | Author | March 2, 2023 | CS |
| Professor Strauss | Reviewer |  |  |
|  |  |  |  |
|  |  |  |  |

**REVISION LEVEL**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
| March 2, 2023 | Version 1.0 | Initial Release |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[Version 1.0 1](#_Toc128698977)

[REVIEW AND APPROVALS 2](#_Toc128698978)

[1.INTRODUCTION 1](#_Toc128698979)

[1.1 Purpose 1](#_Toc128698980)

[1.2 Scope 1](#_Toc128698981)

[1.3 Identification 2](#_Toc128698982)

[1.4 Document Summary 2](#_Toc128698983)

[1.5 System Overview 4](#_Toc128698984)

[2.REFERENCE DOCUMENTS 4](#_Toc128698985)

[3.SYSTEM WIDE DESIGN DECISIONS 4](#_Toc128698986)

[3.1 Software Component Architectural Design 4](#_Toc128698987)

[3.2 Software Architecture General Description 5](#_Toc128698988)

[3.3 Software Item Components 5](#_Toc128698989)

[3.4 Component Interface Identification 6](#_Toc128698990)

[3.5 Software Component Concept of Execution 6](#_Toc128698991)

[4.SOFTWARE ITEM DETAILED DESIGN 6](#_Toc128698992)

[4.1 Structure 6](#_Toc128698993)

[4.2 Static Relationship of Software Unit 7](#_Toc128698994)

[4.3 Behavior 8](#_Toc128698995)

[4.4 Concept of Execution 15](#_Toc128698996)

[4.5 Interface Design 15](#_Toc128698997)

[5.IMPLEMENTATION ARCHITECTURE (NOT REQUIRED) 16](#_Toc128698998)

[5.1 All Active and Passive Classes Assigned to Components 16](#_Toc128698999)

[5.2 Diagrams of Physical Packaging of Logical Components 16](#_Toc128699000)

[6.DEPLOYMENT ARCHITECTURE 16](#_Toc128699001)

[6.1 Physical Deployment Architecture Diagram 16](#_Toc128699002)

[7.DICTIONARIES 17](#_Toc128699003)

[8.](#_Toc128699004)[SOFTWARE ITEM COMPUTER RESOURCE UTILIZATION 20](#_Toc128699004)

[9.REQUIREMENTS TRACEABILITY 20](#_Toc128699005)

[9.1 Software Component-Level Requirements Traceability 20](#_Toc128699006)

[10.SYSTEM DESIGN TESTING 20](#_Toc128699007)

[11.RATIONALE 21](#_Toc128699008)

[12.NOTES 22](#_Toc128699009)

[13.APPENDICES 22](#_Toc128699010)

[13.1 Dictionaries 22](#_Toc128699011)

[13.2 UML diagrams 22](#_Toc128699012)

[13.3 Requirements diagrams 23](#_Toc128699013)

[13.4 Schedule Tracking 25](#_Toc128699014)

[13.4 Defect Tracking 27](#_Toc128699015)

[13.5 Project Schedule 28](#_Toc128699016)

# INTRODUCTION

## Purpose

The Software Design Description documents the system architecture and detailed technical design of the system of the Give&Take project. This document includes the introduction of the project, relative diagrams, specific techniques that will be used in the project, explanation of the architecture, dictionary and traceability of the project, and references of the project. Description shall be used to communicate overall quantitative and qualitative system design and characteristics to operations management, technical design support, training, and operators.

The audience of the document include the technical team of the project, operations manager, software developers, and maintenance team.

## 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

The system of the project is named to be the Give&Take System.

Release Number: SDD-001

## Document Summary

**Title**: Software Design Description (SDD)

Reviewed by Professor Strauss

**Revision Level**: Spring 2023, Version 1.0, Initial Release

**Preface**: This SDD is intended to record the design for the Given&Take project, which is a senior design project at NYU Tandon, Spring 2023.

**Table of Contents**

**Figures:**

Class Diagram

Class Relationship Diagram

Thread Queuing Mechanism

Sequence Diagrams

Collaboration Diagram

Activity Diagrams

State Diagrams

Event Diagrams

Interface Diagram

Physical Deployment Architecture Diagram

## System Overview

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.

# REFERENCE DOCUMENTS

Team B13, Give&Take, Project Proposal, Version 3.0, 02/01/2023

Team B13, Give&Take, SRS-Requirements, Version 3.0, 02/09/2023

Team B13, Give&Take, Software Project Management Plan, Version 2.0, 03/03/2023

# SYSTEM WIDE DESIGN DECISIONS

## Software Component Architectural Design

**Diagram

Description automatically generated**

## Software Architecture General Description

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.

## Software Item Components

In the webpage of Give&Take, the major and only components are user and administrator homepages: each serves for the target group of users without any cross-interaction. Both components contain specific functionalities for their corresponding group of users. For example, the user homepage allows users to make transactions; and the administrator homepage allows administrators to supervise all transactions.

Both homepages connect to the same database, which stores all information from the website such as the transaction details and users’ information. Even though this project is done on campus at NYU, the team does not have the ability to use NYU’s database. Therefore, the team will self-create the database and use it in the project. For example, the user object class in the database will have several attributes, including name, gender, height, weight, amount of credits available, etc.

## Component Interface Identification

* DonateClothes\_to\_Database
* RequestClothes\_to\_Database
* RegisterAccount\_to\_Database
* RateOrder\_to\_Database
* ListItem\_to\_Database
* UserLogin\_to\_Server
* AdministratorLogin\_to\_Server

## Software Component Concept of Execution

The Database component should be launched at first, because this component contains all information required to boost the system. Then, the backend server should be launched and test the basic functionalities of the system as well as the connection with the database. Finally, the frontend webpages should be launched towards the users and administrators of the system. Users and administrators can make updates, and all updates will be recorded in the database.

# SOFTWARE ITEM DETAILED DESIGN

## Structure

* + 1. ***Software Unit Detailed Design***

Graphical user interface, text

Description automatically generated

## Static Relationship of Software Unit

Diagram

Description automatically generated

* + 1. ***Run-time Object Instances***











## Behavior

* + 1. ***Sequence Interaction Diagrams***

A picture containing chart

Description automatically generated

A picture containing timeline

Description automatically generated

A picture containing timeline

Description automatically generated

A picture containing diagram

Description automatically generated

A picture containing timeline

Description automatically generated

A picture containing chart

Description automatically generated

A picture containing chart

Description automatically generated

* + 1. ***Collaboration Diagrams***

Graphical user interface, application

Description automatically generated

* + 1. ***Activity Diagrams***

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

A picture containing shape

Description automatically generated

A picture containing arrow

Description automatically generated

A picture containing text

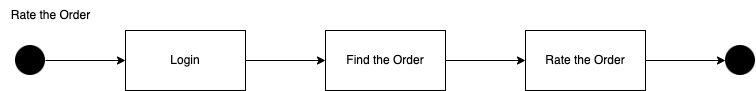
Description automatically generated

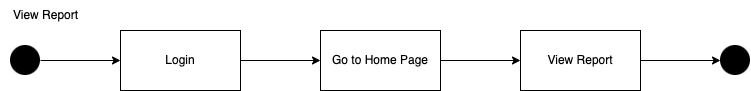
A picture containing shape

Description automatically generated

A picture containing shape

Description automatically generated





* + 1. ***State Diagrams***

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

A picture containing Teams

Description automatically generated

A picture containing Teams

Description automatically generated

A picture containing Teams

Description automatically generated

* + 1. ***Event Diagrams***

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

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with medium confidence

## Concept of Execution

The database shall return the correct information when the user is searching for an

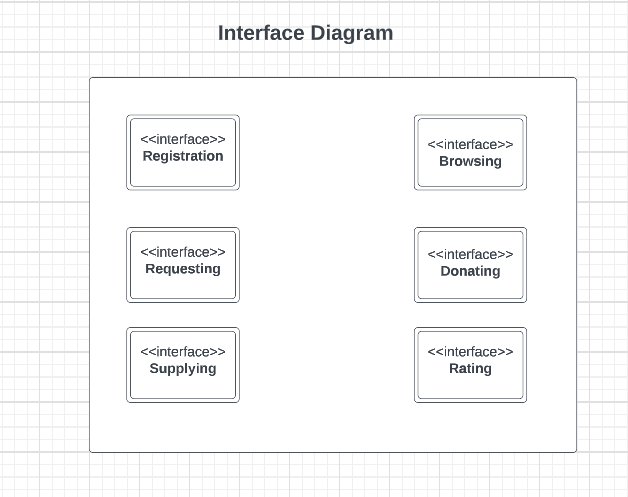
item. The credits shall be deducted according to the final price of the item after every successful purchase. The correct amount of credits shall be added to the supplier once the item is confirmed by the requester. Once a review is posted by the requester, all public users shall be able to see it. All change of user information, successful and unsuccessful transactions, newly added reviews shall be stored and updated in the database.

## Interface Design

* + 1. ***Unique identifier of Interface***

The users of Give & Take, including requester and supplier, shall be able to perform the proper transactions. The supplier shall list their clothing, and the requester shall purchase these listed clothes with the system credits. Once the requester successfully check out, the supplier shall be able to send the item to the requester according to the information stored in the system. The supplier shall donate their clothes to some charity groups through the donation function. The requester shall be able to leave comments to the supplier through the rating function once they confirm the transaction, and all users shall be able to see the review.

* + 1. ***Interface Diagrams***



# IMPLEMENTATION ARCHITECTURE (NOT REQUIRED)

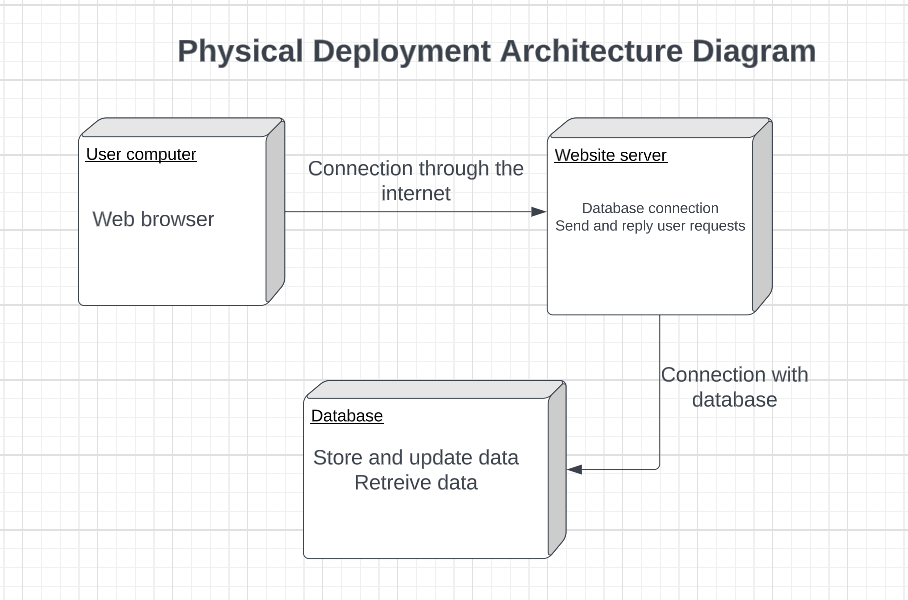
## All Active and Passive Classes Assigned to Components

Includes all files (.CPP, Header, DLL, EXE, etc.) and middleware

## Diagrams of Physical Packaging of Logical Components

# DEPLOYMENT ARCHITECTURE

## Physical Deployment Architecture Diagram



# DICTIONARIES

**Class**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Methods | Attributes |
| Receiver / Donator | User of the website. Users can trade on the website freely. They can either donate or look for free items. | Register as a user of the website; “buy” / “sell” clothes; upload / search for items. | Username; userid; email; address; payment methods |
| Admin | Admin of the website. The admin can delete some irrelevant or harmful content. | Delete item; deal with reports | Adminid; email; adminname |
| Credit | A system which shows the loyalty of the user. Higher credit means more completed trades. | Add credit; deduct credit; user credit / credit ranking | Username; userid |
| Rating | Users can rate each other after every trade. | Leave comment; rate | Username' userid |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Class | Arguments |
| Register as a user | Described in table class | Receiver / Donator | N/A |
| Buy | Described in table class | Receiver / Donator | Click to buy an item from the website |
| Upload / sell items | Described in table class | Receiver / Donator / Admin | Click to upload one or more items on the website for sale |
| Search for items | Described in table class | Receiver / Donator | The type of items a receiver wants to get |
| Add / deduct Credit | Described in table class | Credit | A credit system that will automatically add / deduct points based on each trade |
| Ranking based on credit | Described in table class | Credit | A ranking system based on a user’s credit |
| Commenting | Described in table class | Rating | Users can leave comment under a completed trade |
| rating | Described in table class | Rating | Users can rate each other after each trade |

**Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Description | Simple / Complex | Type | Size |
| Username | A name for every user | Simple | String | About 25 char |
| Userid | A unique id for every user | Simple | String | About 8 -10 char |
| Address | The address of a user | Simple | String | Don't know yet |
| Email | A unique email for every user | Simple | String | Don’t know yet |
| Payment Methods | A method to pay for the item | Complex | String | Don’t know yet |
| Admin Id | A unique id for the admin of the website | Simple | String | About 8 – 10 char |
| Admin name | Name of the admin | Simple | String | Max 25 char |

**Relationship**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Description | From class | To class | Optional / mandatory | cardinality |
| Credit | Users will receive points after trading | Credit | Receiver / Donator | Mandotory | High |
| Rating / commenting | Users can rate / comment after trading | Receiver / Donator | Rating | Optional | High |

**Key Events**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Motive | Action | Pre-conditions | Post-conditions | State Change |
| Donating Clothes | People can donate unwanted clothes via the website, and people who’s in need can get free clothes. | Fashion industry is one of the major reasons of wasting resources. Recycling clothes can help protect the environment, and also help the poor. | The trading is via Give&Take (the website we are implementing) | The donor left unwanted items at home. No enough storage room. | Donators can help people in need, protect the environment, and also clean up their room. | The ownership of the clothes changed between 2 people. |

# SOFTWARE ITEM COMPUTER RESOURCE UTILIZATION

We plan to have 4GB random access memory(RAM), and expand the memory once it reaches 80% of the memory. In addition, we need 256GB to store all data in the database, and we will expand the storage once 80% of itself is occupied. For the central processing unit (CPU), we plan to use 6 CPU cores for MacOS and Windows, and 4 CPU cores for Linux. If the CPU is occupied for more than 60%, we will use more CPU cores.

# REQUIREMENTS TRACEABILITY

## Software Component-Level Requirements Traceability

Registration of New Users, Users’ Profile, Browsing and Jump to Different Pages, Interaction between Users, and Ranking and Yearly Reports can be traced forward.

Authentication of Suppliers and Requesters, Clothes Donation and Clothes Condition Check, Clothes Request, and Non-Commercial Transaction can be traced backward.

The traceability of Other Website-Based Activities is based on the specific requirement.

We hope to ensure all requirements can be traced bidirectional.

# SYSTEM DESIGN TESTING

For the initiation of the project, we decided to build a website and an application for people to exchange clothes without spending money. We also support the cloth donation to charities.

We use the incremental development for the interleaved specification, development and validation.

The basic browsing activity is designed and tested first, then the user login and view personal information is designed and tested, next design and test is the listing and requesting activity and the last is the view and give rating activity.

Each work product is expected to be completed within 2 weeks, and each team member is expected to do the unit testing for their own parts before meeting together for the system testing and review. After complete all the work products, the system will be sent to the school testing team for further testing and review.

Major milestones will include the database design and testing, the website design and testing and the request/supply function design and testing. The baseline is the successful running of the browsing and viewing information, login, requesting, supplying and rating.

Each requirement shall be approved by each team member, the instructor and the project testing team.

The termination activity of this project shall be the completion of all documents, the successful work of the system and the acceptance of the functions from the project testing team and clients.

# 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

None.

# APPENDICES

## Dictionaries

The dictionaries are included here or in section 7.

## UML diagrams

UML diagrams are included in the body of this document.

## Requirements diagrams

Diagram

Description automatically generated

Use case description:

1. Browsing: 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.
2. User register: Users shall be able to register their own unique account by entering necessary information
3. Login: The user can log into their account after successfully registered
4. List item: 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
5. Request: 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
6. Give ratings: 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
7. View Report: 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.

## Schedule Tracking

Time (hours)

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual and team) | Estimated | Actual | Difference |
| SRS - Final | Heng Wu | 1.0 | 1.5 | +0.5 |
| Zhaoyan Zhu | 2.0 | 2.0 | 0 |
| Jin Zhou | 1.5 | 2.0 | +0.5 |
| Carrie Song | 3.0 | 2.0 | -1.0 |
| Team Summary | 7.5 | 7.5 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual and team) | Estimated | Actual | Difference |
| SDD - Initial | Heng Wu | 3.0 | 1.5 | -1.5 |
| Zhaoyan Zhu | 2.0 | 3.0 | +1.0 |
| Jin Zhou | 1.5 | 2.0 | +0.5 |
| Carrie Song | 3.0 | 3.0 | 0 |
| Team Summary | 9.5 | 9.5 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual and team) | Estimated | Actual | Difference |
| SDD Final | Individual members |  |  |  |
|  |  |  |  |  |
|  | Team summary |  |  |  |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual and Team) | Estimated | Actual | Difference |
| Heng Wu | 10.0 | 8.5 | -1.5 |
| Zhaoyan Zhu | 12.0 | 14.0 | +2.0 |
| Jin Zhou | 11.5 | 11.5 | 0 |
| Carrie Song | 14.5 | 11.5 | -3.0 |
| Team summary | 48.0 | 45.5 | -2.5 |

## Defect Tracking

**Defect Counts**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual and team) | Estimated | Actual | Difference |
| SRS - Final | Heng Wu | 2 | 2 | 0 |
| Zhaoyan Zhu | 3 | 2 | -1 |
| Jin Zhou | 3 | 3 | 0 |
| Carrie Song | 2 | 3 | +1 |
| Team summary | 10 | 10 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual and team) | Estimated | Actual | Difference |
| SDD - Initial | Heng Wu | 3 | 2 | -1 |
| Zhaoyan Zhu | 3 | 3 | 0 |
| Jin Zhou | 2 | 3 | +1 |
| Carrie Song | 2 | 4 | +2 |
| Team summary | 10 | 12 | +2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual and team) | Estimated | Actual | Difference |
| SDD - Final | Individual members |  |  |  |
|  |  |  |  |  |
|  | Team summary |  |  |  |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual and team) | Estimated | Actual | Difference |
| Heng Wu | 19 | 15 | -4 |
| Zhaoyan Zhu | 10 | 11 | +1 |
| Jin Zhou | 12 | 15 | +3 |
| Carrie Song | 11 | 16 | +5 |
| Team summary | 52 | 47 | +5 |

## Project Schedule

* + - Gantt or Microsoft Project Schedule

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| WBS | Activity | Task | Assigned | Start Date | End Date | Duration | Status |
|  |  |  |  |  |  |  |  |
| 1.0 | Project Selection |  | Team | January 23, 2023 | February 1, 2023 | 10d | Completed |
|  |  |  |  |  |  |  |  |
| 1.1 | Risk assessment |  | Zhaoyan Zhu | February 1, 2023 | February 1, 2023 | 1d | Completed |
|  |  |  |  |  |  |  |  |
| 1.2 | Project Proposal |  | Team | February 2, 2023 | February 8, 2023 | 7d | Completed |
|  |  |  |  |  |  |  |  |
| 1.3 | Project Information/Description Form |  | Team | February 9, 2023 | February 13, 2023 | 5d | Completed |
|  |  |  |  |  |  |  |  |
| 2.0 | SRS-Complete |  | Team | February 2, 2023 | February 9, 2023 | 8d | Completed |
|  |  |  |  |  |  |  |  |
| 3.0 | SPMP |  | Team | February 13, 2023 | March 3, 2023 | 19d | Completed |
|  |  |  |  |  |  |  |  |
| 4.0 | SDD-Initial |  | Team | February 13, 2023 | March 3, 2023 | 19d | Completed |
|  |  |  |  |  |  |  |  |