Group 10 - Deliverable #1

SFWRENG 3A04: Software Design III - Large System Design

Andrew Hum 400138826
Arkin Modi 400142497
Hongzhao Tan 400136957
Christopher Vishnu 400129743
Shengchen Zhou 400050783

Feburary 7, 2020

1 Introduction

This is the Software Requirement Specification for Hacker Sim.

1.1 Purpose

The purpose of the document is to lay out a description and intention of the software that is being developed for both the client and the developers. It will convey the software and hardware requirements of the game, as well as the design constraints of the software. Furthermore, this document is used to define our product, its functionality and performance which will be used to verify the product in the testing process.

1.2 Scope

The purpose of the proposed software, Hacker Sim, will be an enjoyable, interactive game that will allow the user to grow their specimen and beat their highscores. The focus of the application is to grow a Software Engineer in their room. Both the Software Engineer and their room can be upgraded using the in-game currency. The in-game currency will be acquired through the completion of tasks. Online capabilities will give users the ability to share their rooms and chat as well as give gifts to other users to help complete tasks.

1.3 Definitions, Acronyms, and Abbreviations

SE - Software Engineer

1.4 References

Dormehl, L. (2019, May 29). The Tamagotchi Effect: How Digital Pets Shaped The Way We Use Technology. Retrieved from https://www.digitaltrends.com/cool-tech/how-tamagotchi-shaped-tech/

Technologies, U. (n.d.). Unity. Retrieved from https://unity.com/

1.5 Overview

The document is organized in the following manner: Overall Description, Use Case Diagram, Functional Requirements, Non-Functional Requirements. Overall Description focuses on providing details about how the ideas link together to enable specific functionality and showing functionality through comparison as well as outline who the users are intended to be and the constraints. Use Case Diagrams demonstrate how the stakeholder carries out business events. Functional Requirements explains the functionality of the system. Finally, the Non-Functional Requirements focuses on the required qualities of the software.

2 Overall Description

2.1 Product Perspective

The product is a system modeling application for desktop computers based on the Unity platform using C#. The product design itself is not a continuation of any predecessor products, but in the past there were many implementations with a similar concept; raising a digital being. The trend of virtual pet games started from mid 1990s to early 2000s, popularized by the Tamagotchis. Our product will differentiate itself by giving users an experience of raising a software engineer from the very beginning. Our product is a self-contained system that will not be interacting with other systems.

The system's main features:

1) A server with functionalities:

    - To provide users with online access via login/logout

- To connect a database with the implementation of a game
- To allow different users to share and interact with each other
- To allow users to give feedback and report bugs

2) A database or a file management system on the user's computer for storing:

- User account information set
- User game progress
- Product FAQs for convenient look-up

2.2 Product Functions

2.2.1 Function Summary

The system's functionality is centralized around the user raising their digital specimen as a Software Engineer (referred to as SE below). The user will be involved with the lifecycle of specimen and attempt to promote it with a set of means and reach end-game goal. A currency system is designed for the in-game shop and the user can earn currency from initiating interactions/events with their SE. For example, working and completing coding projects. The in-game shop contains three main categories of items: interactable items (exercise equipment), furniture and room upgrades and food. The specimen has a list of metrics, i.e. tiredness, happiness, age, health, etc. The user must follow certain rules to keep metrics within a rational range, otherwise it will hinder specimen growth or lead to game failure. The user will be able to view their specimen's information to check their progress and current health.

2.2.2 List of Basic Functions

1. User creates a SE.
2. Users initiate events and interact with their SE.
3. Users purchase items from the in-game shop.
4. Users utilize items.
5. Users customize specimen's room.
6. Users view specimen's information.
7. Users reset their game stage and start a new one.

2.2.3 Function Domain

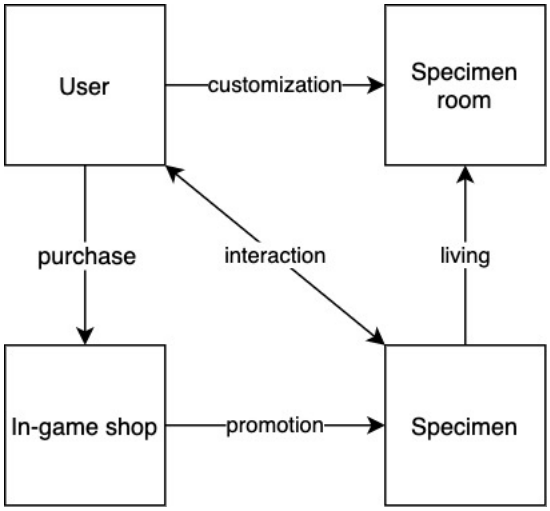


Figure 1: Function Domain

2.3 User Characteristics

Intended users do not have to meet a strict set of requirements to use our product. Users can be a PhD student or no education. They are not required to have past experience of similar types of games either. In general, it is considered that the user should:

- Have a hardware to run our product on, i.e. desktop, laptop, etc.
- Have basic knowledge of operations for computer with access to internet
- Be able to read FAQs and understand how to use the product

2.4 Constraints

Items that might limit the developer's options:

- Regulatory policies: Unity has term of service that developers need to follow
- Basic requirements expectation set by the project outline: Our product must satisfy these functionalities. Features that have a conflict with those won't be our options.
- Project deliverable deadlines and schedule: In the given limiting time, we may not be able to develop various complex functions that are not included in project outlines with complex implementations.
- Developer team's experience: Some people are using Unity platform for the first time and have finite experience with creating and designing games.

2.5 Assumptions and Dependencies

- As mentioned above in section 2.3, all requirements are based on the criteria outlined for proper use of the product. If definitions change during development, requirements will change accordingly.
- All requirements are written under the assumption that our product is able to run on users' hardware - the user's hardware supports the technology of our product.
- It is assumed that the product is running in an environment with internet connection.

2.6 Apportioning of Requirements

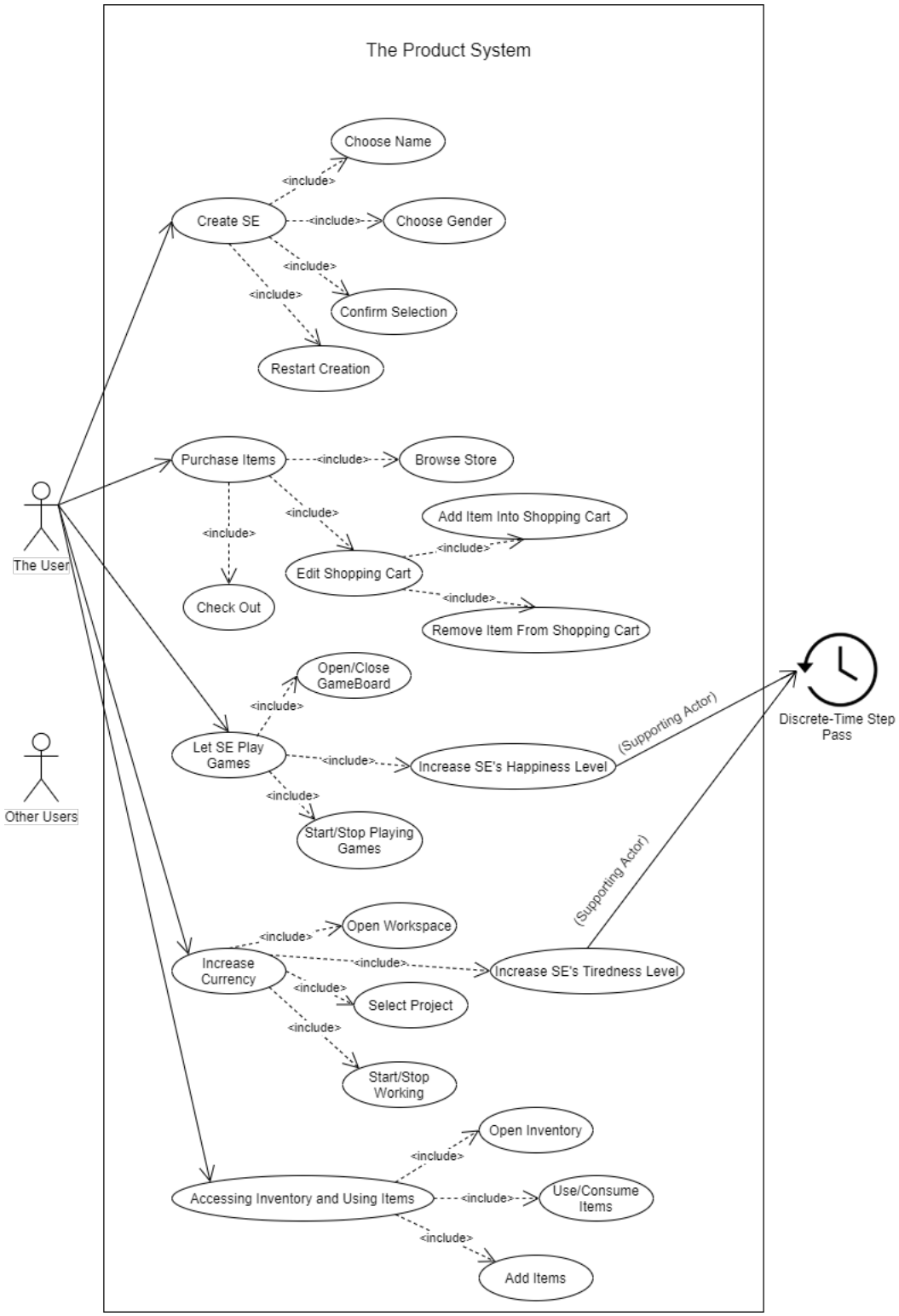
Requirements that may be delayed until future versions of the system:

- Customization of SE's room feature might be delayed as it is not a core feature to support main functions of system and it is not a requirement set by project outline
- Interaction between SEs of different users feature might be delayed if the team cannot find a proper way to achieve this function in the given time

3 Use Case Diagram

In this section, we have three Use Case diagrams to illustrate the required business events and their interactions with each other and the user.

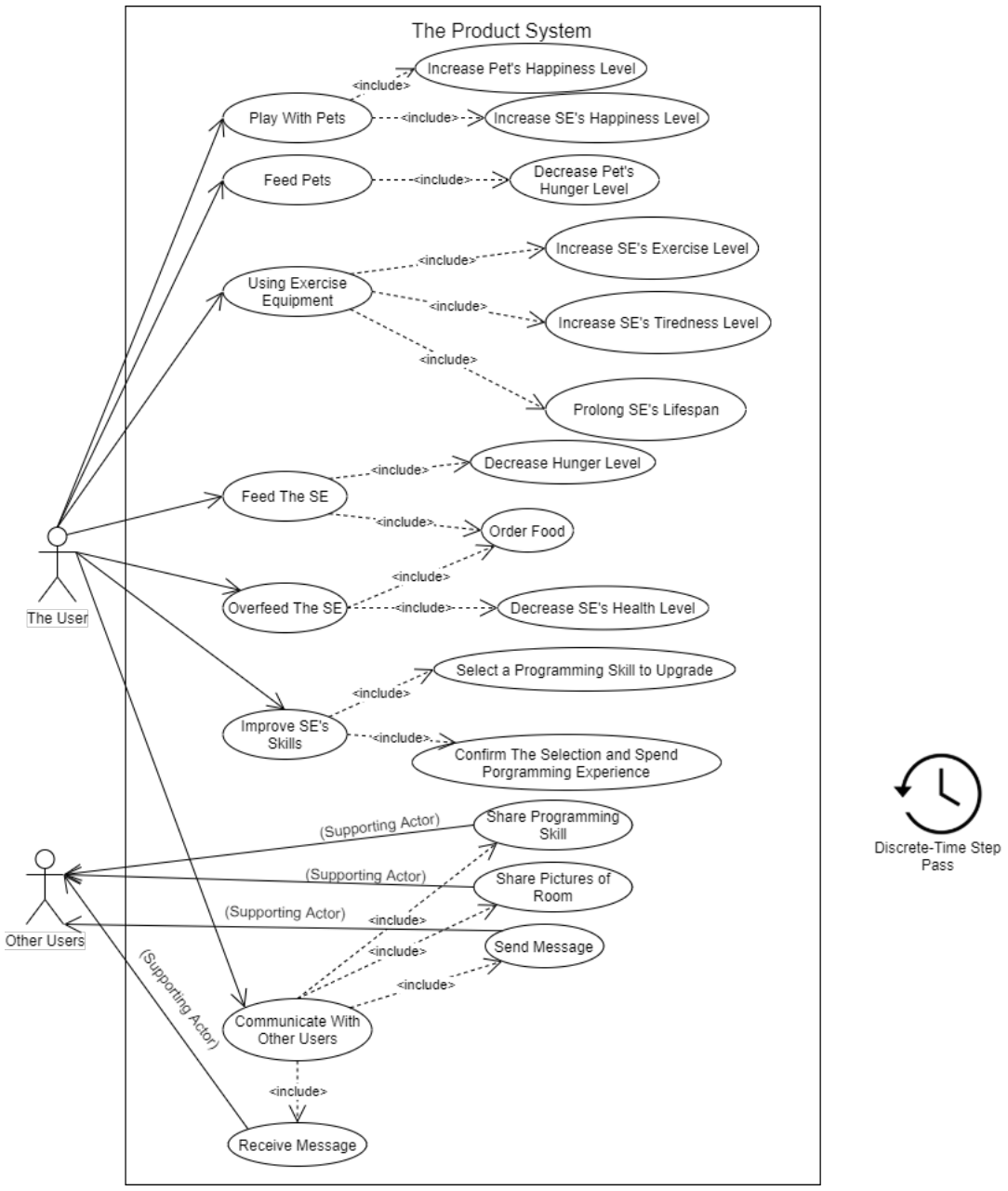
Use Case Diagram 1:



1. Create SE: The user creates his SE, by customizing the basic properties of the SE.
2. Purchase Items: User purchases in-game items for his SE with in-game currency.

3. Let the SE Play Games: The user can let his SE play games to gain happiness.
4. Increase Currency: The user increases his SE's in-game currency by letting SE work on software projects which would also increase the SE's level of tiredness.
5. Accessing Inventory and Using Items: The user accesses the SE's inventory, adds items into inventory and uses the items inside.

Use Case Diagram 2:



1. Using Exercise Equipment: The user lets the SE exercise with its exercise equipment. Letting the SE exercise would keep the SE fit, make the SE tired, and also prolong the SE's lifespan if the exercise equipment have been used frequently.

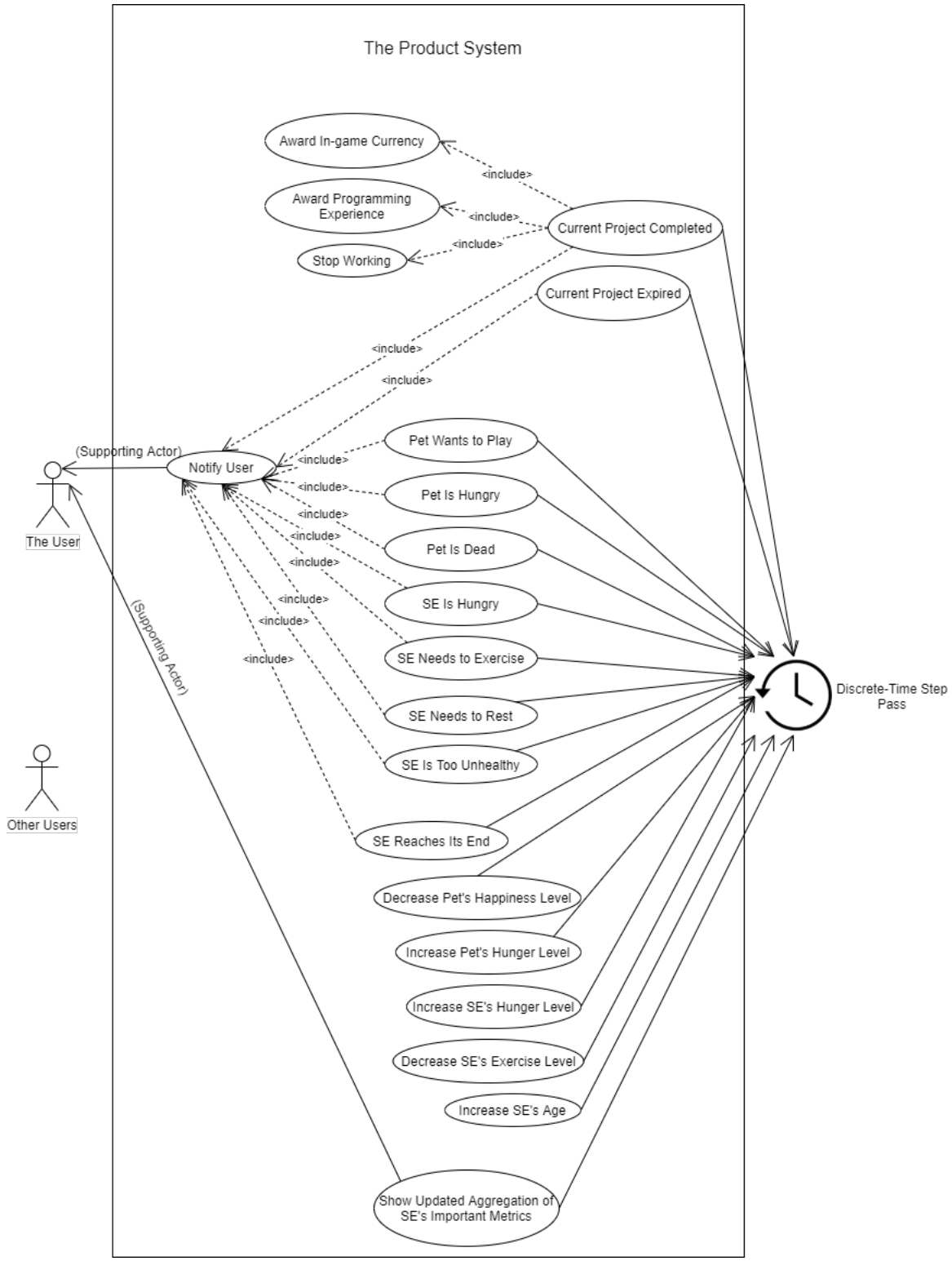
2. Feed The SE: The user shall be able to feed the SE by ordering food in-game. Feeding the SE would decrease the SE's hunger.

3. Overfeed The SE: The user overfeeds the SE by ordering food in-game when the SE is not hungry. Overfeeding the SE could make the SE's unhealthy.

4. Improve SE's skill: The user upgrades the SE's programming skills, by selecting the skill which the user wants to upgrade, confirming his selection and spending certain amount of SE's in-game programming experience to complete upgrading. Upgrading programming skills could increase SE's income.

5. Communicate With Other Users: The user shall be able to communicate through the system with other users. By communicating with other users, the user can send/receive messages to/from other users, share pictures of the SE's room and the SE's current programming skills to other users.

Use Case Diagram 3:



Use-case diagram 3 (references Business Event 5) outlines the interactions between the user and the systems reaction to a discrete amount of time passed. When a discrete amount of time has passed the system will

notify the user of an event that may have occurred. These events include:

- The SE's current state (tired, hungry, etc.)
- Relevant metrics for the SE
- The completion of a current project, and the awarded currency

4 Functional Requirements

The following Business Events are all in the User Viewpoint.

BE1. The user wants to create his SE

1. The system shall allow the user to choose the name of their SE
2. The system shall allow the user to choose the gender of their SE
3. The system shall provide the choice to confirm selection
4. The system shall provide the choice to restart creation

BE2. The user wants to purchase items for the SE

1. The system shall allow the user to browse the in-game store/shop
2. The user shall be able to purchase furniture, exercise equipment and computer upgrades from the shop
3. The system shall provide a choice the user to put/remove items into/from his in-game shopping cart and the total price of the items inside the cart changes correspondingly
4. The system shall allow the user to checkout to purchase all of the items in his in-game shopping cart
5. The system shall check the user's current currency

BE3. The user wants to increase SE's in-game currency

1. The system shall allow the user to open SE's workspace
2. The system shall provide a choice for the user to select the project he wants his SE to work on.
3. The system shall provide the choice to let SE start working
4. The system shall provide the choice to let SE stop working
5. Working shall increase the SE's tiredness level.

BE4. Discrete-time step pass in game

1. The system shall award the user in-game currency and programming experience as the SE completes projects
2. If the project the SE is currently working on is completed, the system shall let SE stop working and notify the user
3. The system shall notify the user once any project SE has processed on passed its due date
4. The system shall show updated aggregation of SE's important metrics (health, happiness, hunger, age, exercise, tiredness)
5. The system shall increase the SE's hunger level overtime
6. The system shall decrease the SE's exercise level overtime
7. The system shall increase the SE's age overtime
8. The system shall notify the user once the SE hungry

9. The system shall notify the user once the SE needs to exercise
10. The system shall notify the user once the SE needs to rest (SE reaching its maximum tiredness level)
11. The system shall notify the user once the SE is too unhealthy (close to reaching its end of lifeline)
12. The system shall notify the user once the SE reaches its end of lifeline
13. The system shall track the user's lifespan and record a score

BE5. The user wants to access their inventory and use items

1. The system shall allow the user to open the SE's inventory display its items
2. The system shall allow the user to use/consume the items in SE's inventory
3. The system shall provide an option to add items to the SE's inventory

BE6. The user want to interact with objects in the room

1. The system shall allow interaction between the SE and its environment
2. Feeding the pet shall decrease the pet's hunger level
3. The system shall allow the SE to use its exercise equipment
4. Using exercise equipment shall increase SE's exercise level
5. Using exercise equipment shall increase SE's tiredness level
6. Using exercise equipment frequently shall prolong the SE's lifespan
7. The system shall allow the SE to play games on it's computer
8. Playing game shall increase the SE's happiness index

BE7. The user wants to feed the SE

1. The system shall provide an option to order food
2. Feeding the SE will reduce its hunger level
3. Feeding the SE when it is not hungry shall cause it to be overfed
4. Overfeeding the SE will decrease its health level.

BE8. The user wants to improve the SE's skills

1. The system shall allow the user to choose a programming skill of the SE to upgrade
2. Upgrading a certain skill shall cost a certain amount of the SE's programming experience.
3. Upgrading certain skills shall increase the SE's income

BE9. The user wants to communicate with friends

1. The system shall allow the user to send messages to other users
2. The system shall allow the user to receive messages from other users
3. The system shall allow the user to share pictures of their room with other users
4. The system shall allow the user to share their current programming skills with other users

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF1. The system shall have a minimalistic design.

5.1.2 Style Requirements

N/A

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH1. The user shall face a minimal number of errors.

UH2. The errors shall have a descriptive message.

5.2.2 Personalization and Internationalization Requirements

UH3. The user shall be able to personalize the SE's name and gender.

5.2.3 Learning Requirements

UH4. The user shall be able to learn the game mechanics through a quick tutorial.

5.2.4 Understandability and Politeness Requirements

UH5. The system icons shall be taken from their common usage icons where applicable.

UH6. The user shall only be promoted with a confirmation box when necessary.

5.2.5 Accessibility Requirements

UH7. The system shall have subtitles under each icon.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR1. The system shall respond to a user's input within 5ms.

5.3.2 Safety-Critical Requirements

N/A

5.3.3 Precision or Accuracy Requirements

PR2. The system shall record all numerical data upto 8 decimal places.

5.3.4 Reliability and Availability Requirements

PR3. The system shall be available at all times except for 2 hours a week for maintenance.

5.3.5 Robustness or Fault-Tolerance Requirements

PR4. When network connection during multiplayer is lost, the system will switch to single-player.

5.3.6 Capacity Requirements

PR5. The system will allow two users to interact via an online session.

PR6. The system will store the player data.

5.3.7 Scalability or Extensibility Requirements

N/A

5.3.8 Longevity Requirements

PR7. The system shall be built and maintained for the duration of the project ending in April 2020.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

N/A

5.4.2 Requirements for Interfacing with Adjacent Systems

N/A

5.4.3 Productization Requirements

OE1. The system shall be distributed via an online repository.

OE2. The system shall be packaged in an executable file (.exe and .app).

5.4.4 Release Requirements

OE3. The system shall have one release at the beginning of April 2020.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

N/A

5.5.2 Supportability Requirements

MS1. The system shall have a tutorial for explanation of the game.

5.5.3 Adaptability Requirements

N/A

5.6 Security Requirements

5.6.1 Access Requirements

SR1. All project members shall have authorized access to all parts of the system at all times.

5.6.2 Integrity Requirements

N/A

5.6.3 Privacy Requirements

N/A

5.6.4 Audit Requirements

N/A

5.6.5 Immunity Requirements

N/A

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

N/A

5.7.2 Political Requirements

N/A

5.8 Legal Requirements

5.8.1 Compliance Requirements

N/A

5.8.2 Standards Requirements

N/A

A Division of Labour

Table 1: Division of Labour

Section	Contributor(s)	Description
Brainstorming	All Members	Developed an idea for the project and answered the minimum requirement questions
Introduction	Christopher Vishnu	Completed and revised the introduction
Overall Description	Shengchen Zhou, Andrew Hum	Shengchen completed all the sections in the Overall Description. Andrew revised the sections and checked for grammar and consistency.
Functional Requirements	Andrew Hum, Hongzhao Tan	Both contributors developed and revised all the Business Events and Functional Requirements
Non-Functional Requirements	Arkin Modi	Completed and revised all the non-functional requirements

Andrew Hum

Arkin Modi

Hongzhao Tan

Christopher Vishnu

Shengchen Zhou