

Software Requirements Specification for DayGame

Yorgos Basioukas, Kapoutselis Christos, Moschopoulos Apostolis,
Papadopoulou Athanasia, Sarafoglou Marina, Spiridopoulos Kostantinos,
Dadidis Mitrofanos, Tsirpanis Theodoris

Chapter 1

Group 7

<i>Name</i>	<i>ID</i>
Basileiadis Anestis	dai19272
Kapoutselis Christos	it14234
Moschopoulos Apostolis	dai19104
Basioukas Yorgos	dai19174
Dadidis Mitrofanos	dai19011
Papadopoulos Dimitris	dai17096
Papadopoulou Athanasia	dai19091
Sarafoglou Marina	dai19080
Spiridopoulos Kostantinos	dai19106
Tsirpanis Theodoris	dai19090

Instructor: Apostolos Ampatzoglou, Alexander Chatzigeorgiou
Course: Software Engineering

Contents

1 Group 7	1
Contents	2
Revision history	3
1. Introduction	4
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Intended Audience	4
1.4 Product Scope	4
1.5 References	5
1.6 Definitions, Acronyms and Abbreviations	6
2. Overall Description	8
2.1 Product Perspective	8
2.2 Product Functions	8
2.3 User Classes and Characteristics	12
2.4 Operating Environment	12
2.5 Design and implementation constraints	12
2.6 Assumptions and Dependencies	12
3. Specific Requirements	13
3.1 Functional Requirements	13
3.1.1 Boss Battle	13
3.1.2 Shop	14
3.1.3 Update Task	16
3.2 Non-Functional Requirements	17
A. Appendix	18
A.1 Product UIs and UCs	18
A.1.1 UIs	18
A.1.2 UCs	21
A.2 Product in a Box	22

Revision history

Name	Date	Reason	Version
Yorgos Basioukas	14/3/2020	Created Document	0.1
Yorgos Basioukas	14/3/2020	Added Table of Contents and Revision	0.2
Kapoutselis Christos Moschopoulos Apostolis Papadopoulou Athanasia Sarafoglou Marina Spiridopoulos Konstantinos Tsirpanis Theodoris	17/3/2020	Added Purpose, Document Conventions, Intended Audience, Product Scope, and Overall Description Section	0.3
Moschopoulos Apostolis Yorgos Basioukas Papadopoulou Athanasia Sarafoglou Marina Spiridopoulos Konstantinos	18/3/2020	Added Functional Requirements	0.4
Kapoutselis Christos Moschopoulos Apostolis Yorgos Basioukas Dadidis Mitrofanos Papadopoulou Athanasia Sarafoglou Marina	20/3/2020	Added Shop and Update Task Use Cases	0.5
Yorgos Basioukas Papadopoulou Athanasia	24/3/2020	Added Boss Battle Use Case	0.6
Kapoutselis Christos Moschopoulos Apostolis Yorgos Basioukas Papadopoulou Athanasia Sarafoglou Marina Spiridopoulos Konstantinos	26/3/2020	Reviewed Use Cases	0.7
Kapoutselis Christos Moschopoulos Apostolis Yorgos Basioukas	26/3/2020	Added Non-Functional Requirements	0.8
Kapoutselis Christos Moschopoulos Apostolis Yorgos Basioukas Papadopoulou Athanasia Sarafoglou Marina Spiridopoulos Konstantinos	27/3/2020	Added References, Definitions, Acronyms and Abbreviations, Product Functions and Appendix	0.9
Kapoutselis Christos Moschopoulos Apostolis Yorgos Basioukas Papadopoulou Athanasia Sarafoglou Marina	29/3/2020	Added Product in a Box, Use Case Diagram, User-Goal Diagram	1.0
Yorgos Basioukas	19/4/2020	Created LaTeX version of this document	1.1

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the functional and non-functional requirements of the desktop application DayGame. It will explain the purpose and features of the software, the interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for users of the software and also potential developers.

1.2 Document Conventions

This Document was created based on the IEEE standard 830 for System Requirement Specification Documents.

1.3 Intended Audience

This document is intended to be used by members of the project team that will implement and verify the correct functioning of the system.

1.4 Product Scope

DayGame is a desktop application that focuses on managing tasks in an interactive way using elements from role-playing games, while tracking everyday tasks and motivating users to complete them in exchange for virtual rewards.

1.5 References

IEEE Recommended Practice for Software Requirements Specifications:

<http://www.cse.msu.edu/~cse870/IEEEExplore-SRS-template.pdf>

IEEE SRS summary and explanation

<https://www.cin.ufpe.br/~if716/arquivos20162/03-IEEE-830>

DayGame's GitHub page:

<https://github.com/teo-tsirpanis/DayGame>

MIT License :

<https://opensource.org/licenses/MIT>

SRS template and Use Case tables provided by instructors

Software development with the use of ICONIX methodology (Greek)

http://users.uom.gr/~achat/AdvSoftEng/ICONIX_eBook.pdf

Software Requirements Specification for Gephi

https://gephi.org/users/gephi_srs_document.pdf

Software Requirements Specification for TimeTracker 2.0

<https://trello-attachments.s3.amazonaws.com/58c0dc2106266ad55c4a3486/58c65f5a171e24879e1e0d02/>

Software Requirements Specification for AASTU Digital Information Desk

<https://www.studocu.com/row/document/addis-ababa-university/software-engineering/mandatory-assignments/ieee-830-1998-standard-srs-document/1981874/view>

Software requirements specification Wikipedia

https://en.wikipedia.org/wiki/Software_requirements_specification

1.6 Definitions, Acronyms and Abbreviations

RPG: Role Playing Game

UI: User Interface

UC: Use Case

FAQ: Frequently Asked Questions

Actor: A user whose defined user goal and is fulfilled by the *System*.

Appearance: Depending on *Gender*, *Appearance* is a visual representation of the *Character* or the *Boss*.

Armor: *Item* that adds *Defence* when *Equipped*.

Attack: *Damage* being done from the *Character* or the *Boss*.

Bag: Stores *Consumable Items* from the *Inventory* that the *Character* can use in the *Boss Battles*. There is limited space. Can not be changed during the *Boss Battles*.

Boss: A *System*-controlled enemy.

Boss Battle: A digital fight between a *Boss* and a *Character*.

Character: An *Actor*.

Complete Tasks: Finished *Tasks*. *Character* wins *in-game balance* and *experience points*. For “*Dailies*” or “*To-Dos*” , the system also checks whether the *Due Date* set by the user has not expired.

Consumable Items: *Items* that can be used once. There are two types. *Spells* and *Potions*.

Damage: It refers to an integer amount of *Hit* or *Life points* being lost.

Defence: It refers to an integer amount of points that is negated from *Damage* dealt by a *Boss*.

Difficulty: There are four types. Easy, Medium, Hard, DARK SOULS used in creating a *Task* with different rewards in *Experience Points* and *In-game balance* when *Completed*.

Due Date: Only applicable to “*Dailies*” and “*To-Dos*” . The date is specified by a number in the calendar.

Entities: The *Character* and the *Boss*.

Equip Items: User can *Equip Items* to the *Character* that increases his *Stats*, in order to have greater chances of winning in *Boss Battles*. Can not be done during the *Boss Battle*. There are limited slots, for example 1 for *Weapon* and 1 for *Armor*.

Equippable Items: Purchased *Items* that can be *Equipped* to the *Character*. There are two types. *Weapons* and *Armor*.

Experience Bar: The visual representation of *Experience Points* in a bar.

Experience points: Indicates the amount of integer points achieved by *Completing Tasks* or winning *Boss Battles*. Reaching a specific amount of points increases the *Character's Level*. After the *character Levels Up* the points are reset to zero.

Frequently Asked Questions: A set of questions and answers made by the developers for easier program use and problem solving.

Gender: Male and Female options.

Hit points: *Hit points* are used only in *Boss Battles*

- For *Characters*: A copy of current *Life points*
- For *Bosses*: An integer amount of points that indicates the *Boss*' continued ability to function during the *battle*

Hit point regain: An integer amount of points added to *Hit points*

In-game balance: Digital currency that can be spend on the *Shop*. It can be acquired through *Completing Tasks* or *Boss Battles*.

Items: A set of digital products that can be purchased from the *Shop* with *In-game balance*. Different kind of *Items* have different *stats*.

Inventory: Placeholder for all *Items* purchased at the *Shop*.

Item category: Armor, Weapons, Spells and Potions.

Jackpot: Earning double amount of *rewards* after completing a *task*.

Level: An integer number that shows the rate of game progression and difficulty

Level up: When the character gets all required *Experience Points* it *Levels Up*. With this process it regains all *Life points* and *Experience Points* are reset to zero. In addition you can increase a *Stat* type of *Damage* or *Defence*.

Life Bar: The visual representation of *Life points* in a bar.

Life points: An integer amount of points that indicates the *Character*'s continued ability to function out of *Boss Battles*. The character dies after it reaches 0 points or less, losing all progress.

Luck: A *stat* that defines the possibility to get a *jackpot* after completing a task.

Navigate Menu: The main menu of the program which is always locked in the UI when user interacts with the game.

Procrastinate Tasks: Unfinished *Tasks*. *Character* loses *in-game balance*, *experience points* and *Life points*. For "Dailies" or "To-Dos", the system also checks whether the time frame set by the user has expired

Potions: *Consumable Items* that *regains hit points* to the *character*.

Quest Log: The main page where the user lands after *character* selection or creation. It contains all tasks and their functions.

Rewards: There are two types. classic reward multiplied by one and *jackpot*.

Shop: The place where you can buy *items*.

Spells: *Consumable items* that deal more *Damage* to *Bosses*.

Start Page: The window when the user opens the program

Stats: Types are *Damage*, *Defence*, *Hit points*, *Luck*.

System: A group of interacting or interrelated entities that form a unified whole

Tasks: A *Task* is defined as an achievement to be reached. It is set by the user and falls under the following categories

- Dailies: *Tasks* that check completion on week-days, scheduled by the user
- Habits: *Tasks* that refresh every day
- To-Dos: *Tasks* that check completion on specific days

Turn Based Battle: Turn based battle is where the *Actor* and the *Boss* are playing with turns. The *Actor* is making a move and the *Boss* is waiting for his own.

Unequip Items: User can *unequip items* from the *character* in order to *equip* better *items*. Can not be done during the *Boss Battles*.

Visibility: All *Items* are visible at first. User can select to apply a filter for a specific *item category* that hides the rest of the *Items* not belonging to that category.

Weapon: Item that increases *Damage* dealt to *Bosses* when equipped.

2. Overall Description

2.1 Product Perspective

DayGame is a gamified task management application, that provides users the ability to engage with their everyday tasks and habits. By adding RPG elements, the application intends to add significance to ordinary tasks and thus make them more appealing and interesting. The users can create their own character, select or add tasks, buy items using in-game balance and battle opponents. The product has been developed to run on Windows environments.

2.2 Product Functions

This section describes concisely all the product functions. For a more general description see Appendix A

Start Page:

- Choose Character: User chooses between their created characters. Data shown are the character's level, name and appearance
- Create Character: Opens a form and user chooses name and gender
- Delete Character : Shows a confirmation message and if it is affirmative then it deletes the selected character and shows the start page, else it shows the start page

Navigate Menu:

- Quest Log: Displays the Task Log menu
- Shop: Displays Shop menu
- Inventory: Displays Inventory menu
- Battle: Displays the Boss Battle menu
- Help: Displays FAQ and feedback menu

Character Profile:

- Contains character's data
 - Character Appearance
 - Character Name
 - Experience Bar
 - Life Bar
 - Character Level
 - In Game Balance

Quest Log:

- Task Log: A log containing three types of ongoing tasks: Habits, To-Dos, Dailies
- Add task: User creates a task and it is then stored into the Task Log. User needs to write the name of the task, the Task Type, its Difficulty and the Due Date, if possible
- Update Task
 - Complete Task: Marks a task as completed
 - Procrastinate Task: Marks a task as unfinished
 - Level Up if preconditions are met
- Delete Task: Deletes task from the log
- Drag And Drop Tasks: The ability to switch tasks placement

Shop:

- Item filter: Changes item visibility based on the users choice
- View Item: Opens window that shows the item's stats and description
- Buy Item: Adds item in the user's inventory and updates the user's gold balance

Inventory:

- Item filter: Changes item visibility based on the users choice
- View Item: Opens window that shows the item's stats and description
- Equip Item: Equips the selected item to the character.
- Unequip Item: Unequips the selected item from the character

Boss Battle:

- Turn based battle system
- Selection menu: Attack, Spells, Potions
 - Attack: The user's character performs an attack on the boss
 - Spells: The user's character performs a special attack on the boss using an item
 - Potions: The user's character performs a self-healing ability using an item
- Use Items: uses a consumable item from the bag
- Boss Functions: Attacks after user's turn
- Character deals damage depending on level and items that add to the overall stats
- Boss deals damage depending on character's level

Character Death:

- Shows appropriate message and character loses one level

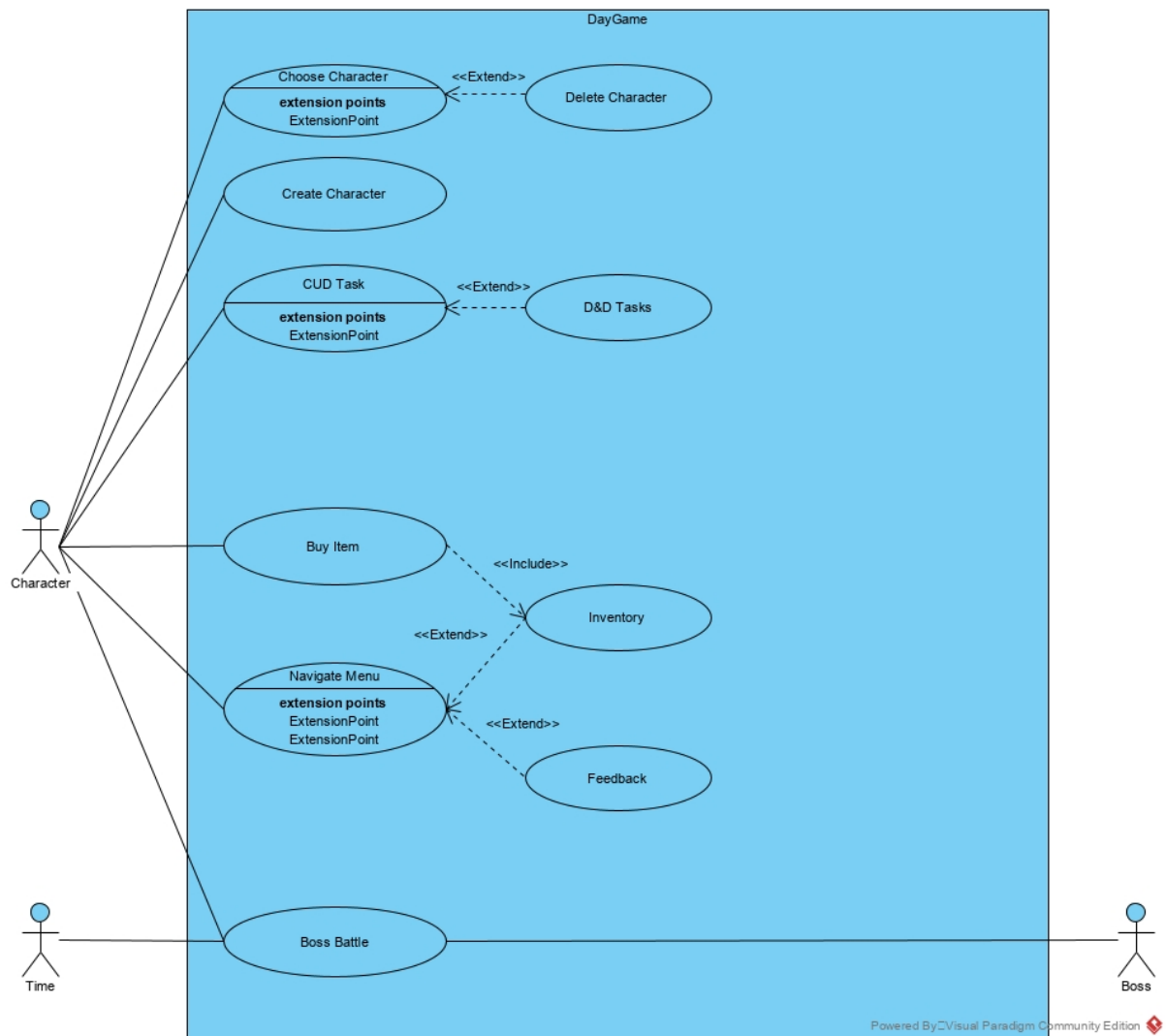
Help:

- Contains a Frequently Asked Questions form
- Feedback: Shows a text field for the user to give feedback or report a bug

Autosave:

- System saves characters' data and progress in a text file.
- Read: every time a user chooses a character, the system loads the appropriate file from the saved log directory
- Write: System writes into a text file all progress done in the game and stores it to a saved log directory

Use Case Diagram:



User-Goal Diagram:

User	Goal
Character	Choose Character Create Character Delete Character Navigate Menu CUD Tasks D&D Tasks Buy Item Inventory Feedback Boss Battle
Boss	Boss Battle
Time	Boss Battle

2.3 User Classes and Characteristics

Anyone that desires to organize their time. There are no specific characteristics required for use.

2.4 Operating Environment

DayGame will be able to run on:

- Windows 7 SP1+ on both architectures (x64, x86)
- Windows 8.1 on both architectures (x64, x86)
- Windows 10 Version 1607+ on both architectures (x64, x86)

2.5 Design and implementation constraints

- System requirements from a casual web browsing computer are enough to run the program
- Quantitatively data are subject to change and sometimes are not defined in this document. Testers will be responsible for this job

2.6 Assumptions and Dependencies

- The users have sufficient knowledge of operating a computer
- The users know the English language, as the user interface will be provided in English

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Boss Battle

Use Case Name	Boss Battle (UC-10)
----------------------	---------------------

Brief Description

The use case is called after the precondition is met to fight a strong boss. User ought to have completed enough tasks by that time, in order to get items to increase their chances of winning. If the user loses, they will have to fight the boss with the same stats again

Flow of Events

Basic Flow

- 1 After the precondition is met. System shows UI-12 “Boss Battle”
- 2 User makes the first move by choosing either “Attack” , “Spells” or “Potions”
- 3 User selects “Attack”
- 4 User deals damage to the boss. System shows message ” +x damage” , x being the number of damage that the user can attack with.
- 5 System updates the Hit points of the boss
- 6 Boss makes second move that deals damage to the user. System shows message ” +x damage” , x being the number of damage that the boss attacks with
- 7 System updates the Hit points of the user
- 8 The UC goes to step 2 of Basic Flow, until one of the entities loses all Hit points
- 9 Boss loses all Hit points. The UC ends. User wins experience points and in-game balance. System calls UI-07 “Quest Log”

Alternative Flow

- 3a1 User selects “Spells” . System shows acquired spells
- 3a2 There are spells in the bag. User selects one spell to damage the boss
- 3a3 The UC goes to step 5 of Basic Flow
- 3a2a There are no spells in the bag. User selects to go back
- 3a2b The UC goes to step 2 of Basic Flow

continued on next page

-
- | | |
|------|--|
| 3b1 | User selects “Potions” . System shows acquired potions |
| 3b2 | There are potions in the bag. User selects one potion to regain Hit points |
| 3b3 | System updates Hit points of user |
| 3b2a | There are no potions in the bag. User selects to go back |
| 3b2b | The UC goes to step 2 of Basic Flow |
| 9a1 | User loses all Hit points. The UC ends. User loses all equipped items and has to repeat the same boss depending the precondition. System calls UI-07 “Quest Log” |

Preconditions

The boss will appear randomly between 2-7 days after the previous battle or the character creation. If there is no previous battle, the boss will appear 4-7 days after the new character is created.

Post-Conditions

The following post-condition is going to be met in the end of UC-10 execution:

- System calls UI-07 “Quest Log”

3.1.2 Shop

Use Case Name

Buy Item (UC-08)

Brief Description

The use case is called when the player wants to buy items for his character. After entering the shop, the player can buy Armor to increase his self defence, Weapons and Spells to increase his damage output and Potions to (re)gain Hit points in boss battles

Flow of Events

Basic Flow

- | | |
|---|--|
| 1 | User selects to open “Shop” |
| 2 | System shows UI-10 “Shop” with all the items, their price tags and their stats |

- 3 User selects a filter of his choice (Armor, Weapon, Potions, Spells) that only shows the items of the selected category
- 4 User selects to buy an item
- 5 If money is enough, system shows message “Purchase completed” and automatically reduces user’s balance and the item is added to his inventory
- 6 User wants to exit the shop. The UC ends. System calls UI-07 “Quest Log”

Alternative Flow

- 3a1 User can also scroll through the list of all items without selecting a filter.
- 3a2 The UC goes to step 4 of Basic Flow
- 5a1 User does not have enough money. User can not buy the item
- 5a2 System shows Error Message “You don’t have enough money to buy this item”
- 5a3 The UC goes to step 2 of Basic Flow
- 6a1 User wants to buy an item again. System calls UC-08 “Buy Item”

Preconditions

There are no preconditions

Post-Conditions

This post-condition is going to be met in the end of UC-08 execution:

- UC-08 “Buy Item” is called to buy an item again
- UI-07 “Quest Log” is called to leave the shop

3.1.3 Update Task

Use Case Name	Update Task (UC-06)
----------------------	---------------------

Brief Description

This use case is called when the user updates existing tasks. There are three types of tasks each with a designated area in the UI: Dailies, Habits, To-dos. User must complete these tasks for in-game survival and real life progression. Update Task is defined as a set of required actions that will specify whether a task has been completed or procrastinated.

Flow of Events

Basic Flow

- 1 When the user is at UI-07 “Quest Log”
- 2 User selects to update a task
- 3 System shows UI-09 “Update Task”
- 4 User selects to complete a task. System rewards the user with experience points and in-game balance depending the task type
- 5 Experience points are not enough to level up. Use case ends. System shows UI-07 “Quest Log”

Alternative Flow

- 4a1 User selects to procrastinate. System punish the user with a loss in Life points, experience points and in-game balance depending the task type
- 4a2 Life points is a non zero, positive number. The UC goes to step 7 of Basic Flow
- 4a2a Life points are zero or negative number. System calls UI-13 “Character Death”
- 5a1 Experience points are enough to level up. System levels up the player, shows message “congratulations you leveled up, choose a stat to increase” and resets experience point bar to 0
- 5a2 User selects a stat to increase. System increases that character stat
- 5a3 The UC goes to step 7 of Basic Flow

Preconditions

One condition must be met in order for UC-06 “Update Task” to execute:

- UI-07 “Quest Log”

continued on next page

Post-Conditions

One of the following post-conditions is going to be met in the end of UC-06 execution:

- The system calls UI-13 “Character Death”
- The system shows UI-07 “Quest Log”

3.2 Non-Functional Requirements

- **Performance**

- Response time: The data system shall not be deteriorated in response time as the number of quests are added by the user increases. Response time seen by users should be on the order of 4 seconds or less
- Loading speed: The data system shall load as quickly as comparable productivity tools on whatever environment it is running in
- Number of users: The system does not sustain online multiplayer functions. However you can create multiple characters

- **Safety**

Users’ data are saved automatically when changes happen, thus all progress is saved in case of program crash or power loss.

- **Security**

There are no security requirements, because any type of user can use it without any additional privileges.

- **Software maintenance**

Based on potential feedback from testers or users, bug appearances and potential balancing issues, updates will be published.

- **Software reliability**

Software has not been tested yet, which means there is unknown probability for failure.

- **Software quality**

DayGame provides the user with very simple features. Due to its easy to use interface it can be accessed by inexperienced users.

A. Appendix

A.1 Product UIs and UCs

A.1.1 UIs

This section of the Appendix describes all the UIs that will be developed with their data

- UI-01 Start Page
 - UI-02 Choose Character (UC-01)
 - * Character Appearance
 - * Character Name
 - * Character Level
 - UI-03 Create Character (UC-02)
 - * Name
 - * Gender Selection Menu
 - UI-04 Delete Character (UC-03)
 - * Character Deletion Confirmation
- UI-05 Navigate Menu (UC-04)
 - Quest Log
 - Shop
 - Battle
 - Inventory
 - Help
- UI-06 Character Profile
 - Character Appearance
 - Character Name
 - Experience Bar
 - Life Bar
 - Character Level
 - In Game Balance

- UI-07 Quest Log
 - To Do Log
 - Dailies Log
 - Habits Log
 - UI-08 Add Task (UC-05)
 - * Name
 - * Task Type
 - * Difficulty
 - * Due Date
 - UI-09 Update Task (UC-06)
 - * Complete Task
 - * Procrastinate Task
 - Delete Task (UC-07)
 - Drag and Drop Tasks (UC-12)
- UI-10 Shop
 - Buy Item (UC-08)
 - Item Filter
 - * Armor
 - * Weapons
 - * Spells
 - * Potions
 - Items List
 - * Item Image
 - * Item Name
 - * Item Stats
 - * Item Price
 - * Buy

- UI-11 Inventory (UC-09)
 - Item Filter
 - * Armor
 - * Weapons
 - * Spells
 - * Potions
 - Acquired Items List
 - * Item Image
 - * Item Name
 - * Item Stats
 - Equip Items
 - Unequip Items
- UI-12 Boss Battle (UC-10)
 - Selection Menu
 - * Attack
 - * Spells
 - * Potions
 - Boss Profile
 - * Boss Name
 - * Boss Hit
 - * Boss Appearance
 - Character Profile
 - * Character Name
 - * Character Hit
 - * Character Appearance
- UI-12 Help
 - Frequently Asked Questions
 - UI-14 Feedback (UC-11)
- UI-13 Character Death
 - You lose one level
- Autosave
 - Read/Write a text file

A.1.2 UCs

This section of the Appendix lists all the UCs that will be developed

- Choose Character UC-01
- Create Character UC-02
- Delete Character UC-03
- Navigate Menu UC-04
- Add Task UC-05
- Update Task UC-06
- Delete Task UC-07
- Buy Item UC-08
- Inventory UC-09
- Boss Battle UC-10
- Feedback UC-11
- D& D Tasks UC-12

A.2 Product in a Box

