Name: John Karstin Neill Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature provides a foundation for all of the main Character’s skills, and score. It maintains and updates these skills to allow the Character to grow and improve over time. It also calculates the current score of the player using its experience, and delivers it to the score server.

## Use case diagram with scenario \_\_/14

### Use Case Diagrams

### 

### Scenarios

**Name:** Set Default Character Data

**Summary:** The Menu tells the system which character is selected, and default skill values are set

**Actors:** Menu System

**Preconditions:** Skill System has been initialized

**Basic sequence:**

**Step 1:** Accept selected character

**Step 2:** Collect selected character data from database

**Step 3:** Set character base skill defaults

**Step 4:** Set skill modifier defaults to zero

**Exceptions:**

**Step 1:** Character is not in character database: use predefined default

**Step 2:** A character has already been selected: reset character with new base skill data

**Post conditions:** Update Skill Data with the given base character data

**Priority:** 1\*

**ID:** C01

**Name:** Update Character Skill Data

**Summary:** Applies given changes to base data, and provides the net Skill state

**Actors:** Menu, Character Control

**Preconditions:** Character and base skills have been initialized

**Basic sequence:**

**Step 1:** Accept Skill changes

**Step 2:** Apply Skill changes to skill modification values

**Step 3:** Recalculate net Skill data

**Exceptions:**

**Step 1:** Changes put Skill values below zero: floor at zero

**Step 2:** Skill to be changed does not exist: ignore change

**Post conditions:** Store net skill result in internal storage

**Priority:** 1\*

**ID:** C02

**Name:** Provide Character Skill Data

**Summary:** System provides a read-only access point to current Skill data

**Actors:** Menu, Character Control

**Preconditions:** Net Skill data has been stored

**Basic sequence:**

**Step 1:** Grab net skill data from internal storage

**Step 2:** Deliver copy of net skill data

**Exceptions:**

**Step 1:** Request for data made before any has been stored: return -1

**Post conditions:** None

**Priority:** 1\*

**ID:** C03

**Name:** Apply Power-Up

**Summary:** Update Skill data for a temporary period, then return to normal

**Actors:** Character Control

**Preconditions:** Character and base skills have been initialized

**Basic sequence:**

**Step 1:** Accept Power-Up skill change info

**Step 2:** Update skill change in modification data

**Step 3:** Wait for time, then remove modification

**Exceptions:**

**Step 1:** Power-Up timer is already running/Power-Up is already in place: ignore request

**Step 2:** Selected skill does not exist: ignore request

**Post conditions:** Allow other Power-Ups to be used

**Priority:** 3\*

**ID:** C04

**Name:** Add XP

**Summary:** Take given experience and apply it to skill data.

**Actors:** Character Control

**Preconditions:** Character and base skills have been initialized

**Basic sequence:**

**Step 1:** Accept experience change

**Step 2:** Apply skill change due to new experience

**Step 3:** Recalculate net skill data

**Exceptions:**

**Step 1:** Experience given is negative or zero: ignore change

**Post conditions:** Update Menu and Character Control about changes

**Priority:** 2\*

**ID:** C05

**Name:** Calculate Score

**Summary:** Calculate and provide the current score to the Score Server

**Actors:** Score Server

**Preconditions:** Character and base skills have been initialized

**Basic sequence:**

**Step 1:** Gather current experience points

**Step 2:** Gather player display name data

**Step 3:** Combine into Server deliverable

**Step 4:** Send data to server

**Exceptions:**

**Step 1:** Connection with Server is not established: discard score calculation

**Step 2:** Server data delivery failed: discard score calculation

**Post conditions:** Clear local version of score to save resources

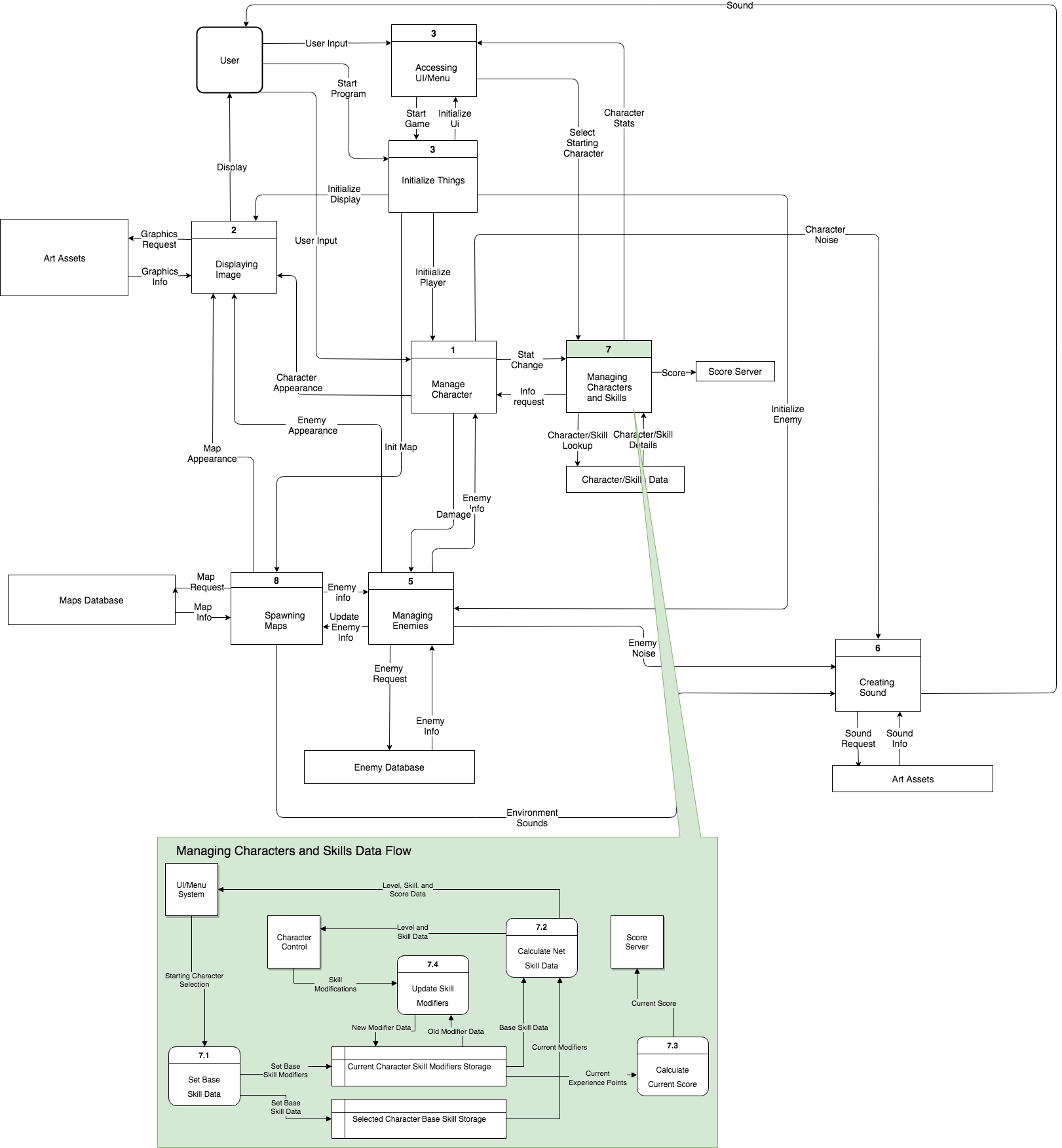
**Priority:** 2\*

**ID:** C06

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_/14

### Data Flow Diagrams



### Process Descriptions

**7.1: Set Base Skill Data**

ACCEPT selected character

IF selected character is not in database

SET selected character to predefined default

GET base data from external database

SEND base data to base data internal storage

SET base modifier data to zeroes

**7.2: Calculate Net Skill Data**

ACCEPT base skill data AND modifier data

SUM base and modifier data

IF any net data is negative after sum

FLOOR data at zero

SEND net skill data to Menu and Character Control

**7.3: Calculate Current Score**

GET current experience data AND name of player

SEND copy of data to the Score Server

**7.4: Update Skill Modifiers**

ACCEPT skill data modification

IF skill does not exist

IGNORE request

ELSE

GET current modification data

ADD new modification to old modification data

SET new modification data

## Acceptance Tests \_\_\_\_\_\_\_\_/9

**Set Base Skill Data feature**

Run 1000 times with random data to ensure that correct information is stored in correct locations.

* Start loop to run 1000 times
* Generate random data and attempt to store data
* Check data to make sure it is stored correctly

**Calculate Net Skill Data feature**

Run 1000 times with random stored data to ensure correct net data is calculated and delivered.

* Start loop to run 1000 times
* Generate random data and store in correct data storage
* Send data through calculation feature
* Check to make sure data sum matches expected result

**Calculate Current Score feature**

Run 1000 times with random experience to ensure that correct score and user information is retrieved to deliver to the server.

* Start loop to run 1000 times
* Generate random data and store in internal storage
* Generate server deliverable
* Check to ensure that deliverable matches what is expected
* Attempt to send to server
* Check server connection handling

**Update Skill Modifiers feature**

Run 1000 times with random data to ensure that skill updates are changed accurately.

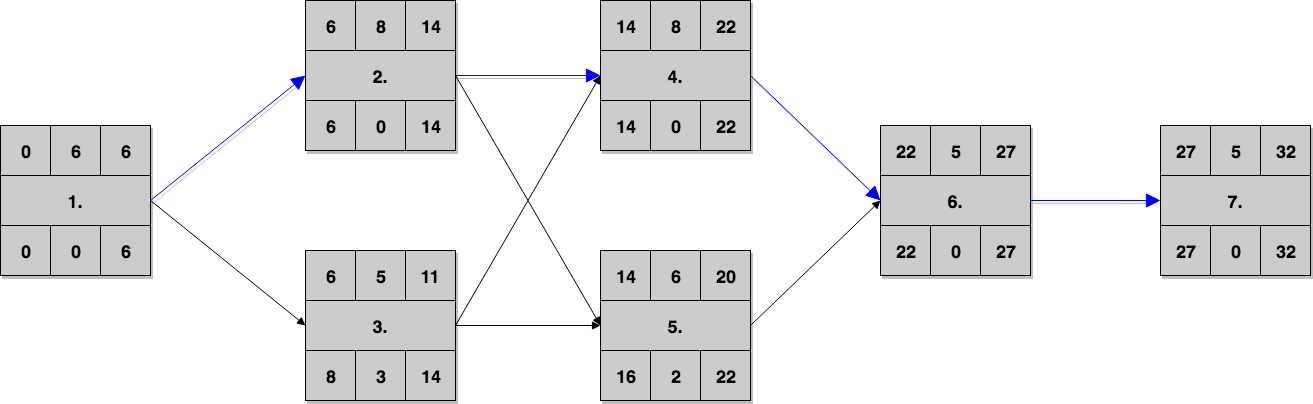
* Generate random data and store in internal storage
* Start loop to run 1000 times
* Generate random change data
* Attempt to send change to data storage
* Check data to ensure change took place as expected

## Timeline \_\_\_\_\_\_\_\_\_/10

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (hrs) | Predecessor Task(s) |
| 1. Research | 6 | - |
| 2. Structure Design | 8 | 1 |
| 3. Database Design | 5 | 1 |
| 4. User Documentation | 8 | 2, 3 |
| 5. Programming | 6 | 2, 3 |
| 6. Testing | 5 | 4, 5 |
| 7. Installation | 5 | 6 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | , | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | , | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 | , | 5 |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |