**Software Implementation and Testing Document**

**For**

**Group 11**

Version 1.0

**Authors**:

Alexis Amoyo

Katelyn Fischer

Hannah Housand

Olivia Mei

Sophia Quinoa

# Programming Languages (5 points)

Ren’Py: The engine that we chose to implement our visual novel in has their own language. As a result, most of the game will be written in Ren’Py scripts, including the story, visual changes, and save options.

Python: We will use Python to create the minigames throughout each of the stories. We chose this because Ren’Py is implemented in Python making integration of the code seamless.

# Platforms, APIs, Databases, and other technologies used (5 points)

Ren’Py: Game engine that is used throughout the entire project to create, run, and deploy the game.

# Execution-based Functional Testing (10 points)

The bulk of our execution-based testing was testing the code for the basic structure of the game. We made sure that we could progress through a story by clicking anywhere on the screen and that we could transition through each chapter, especially because we split up the chapters into separate files of code. We also tested to make sure backgrounds and characters could be properly displayed with the correct dimensions by creating some sample art.

# Execution-based Non-Functional Testing (10 points)

Our non-functional requirements were tested by running a basic shell of what our game will end up being. We tested the platform on both Windows and Mac operating systems and made sure the game ran smoothly. Since the game does not have the bulk of its content, we will need to continually make sure that the non-functional requirements are met. Additionally, when the game runs, some metrics are logged in logs.txt file. This file contains statistics of how long each process took to run and what distribution it is being run on. After each run, we can check this file to ensure that we are fulfilling the non-functional requirements.

# Non-Execution-based Testing (10 points)

For many of the functional requirements, we performed non-execution-based testing to ensure that our plan for the game will have everything that we need and that all group members agree with what components we would like to see in the game. This testing was done by creating a detailed outline of how the functional requirements will be met. Additionally, a review was conducted to check what each person has planned for their chapter of the game and to offer suggestions on where to make improvements.