Software Requirements Specification

for

Structural Health and Rupture Detection (SHARD)

Version 1.0

Prepared by Julian Herrera and Matthew Manley

Florida Institute of Technology

10/5/2022

Table of Contents

1 Introduction

1.1 Purpose

The purpose of the System Requirement Specification document is to outline the main functionalities of the software that will be utilized for the Structural Health and Rupture Detection project. The document will detail the various functional, performance, and usability requirements of the software. The intended audience of this document is Dr. Willard, the main client for the S.H.A.R.D project. Our faculty advisor, Dr. Silaghi, and our senior project professor, Dr. Chan, will also be reviewing this document.

1.2 Product Scope

The overall scope of the SHARD project within all disciplines is that there is a need for a product that protects against micrometer space debris impacts while relaying data from those impacts to the user. The software scope designated for this document is a software that connects with the hardware of the system and graphically displays the physical structure and the data for each impact on the structure. The hardware of the system involves multiple sensor connections (indicating multiple tiles on the physical structure) connecting to a multiplexer, relaying all of the sensor data to a microcontroller which feeds through to the computer.

Integrated into this software also is a testing environment which simulates a user created model withstanding against user generated projectiles launched at the model, testing the health for each tile that is hit by each projectile. The benefit of the simulation environment is that the user can generate test scenarios of different structures in order to better understand the physical structure that the user will be working with.

The benefit of the software as a whole is that it provides the user with a graphical display that gives simple visual (and analytical) feedback on otherwise complicated and convoluted mathematical models of micrometer space debris collisions against a hardened shield, as well as providing the user with a simple yet very important alert system for when there is a life threatening breach in the tiles.

1.3 Definitions, acronyms, and abbreviations

Throughout the functional requirements, we say the terms “physical model” and “simulation mode” frequently. A simulation model is a model that is to be used for simulating purposes only. It becomes active (i.e. a simulation is running) when the user activates it. A physical model is the same concept except it is to be used only for the connection with the hardware and software interface. It becomes active when the user wishes to receive and transmit data to the hardware system.

2 Overall Description

2.1 Product Functions

A major function of the product is for the ability to display three dimensional, virtual models, of either a physical structure or a user generated structure. The product will display a virtual model of a physical structure when the software is connected to specific hardware. The product will display a virtual model of a user generated structure when the software is in the specific mode that allows for this generation. That same mode allows the user to use the model which was created by them to simulate different projectiles launched at the model at varying times. The user can define projectile events, which are events that occur at a specific time in the simulation wherein a user defined projectile is launched at a user defined trajectory towards (or away from) the model.

2.2 User Classes and Characteristics

The intended users for this software are astronauts as that is what the original scope requires. However, this product could also be applied to anything that requires a shell structure and has the potential to be breached. For example, a ship's hull or a car's body. Therefore, the users of this product are any operator or maintainer of a system that has an external shield frame.

2.3 Operating Environment

The product will be compatible with the major operating systems present today (Windows, Mac, Linux).

2.4 Assumptions and Dependencies

We assume that each sensor will be connected to a multiplexer. Each multiplexer will have a maximum of 16 connections, which allows for 8 tiles to be connected at one time (because there are 2 sensors per tile).

2.4 Revision History

Revision Number	Description	Date
1.0	Initial revision	10/5/2022

3 External Interface Requirements

3.1 User Interfaces

[SYS-HLR-01] Must have a mouse.

3.2 Hardware Interfaces

[SYS-HLR-02] There shall be a multiplexor that takes up to 16 sensor inputs from a grid of sensors.

[SYS-HLR-03] The product will have the ability to have more than one multiplexer, allowing for more than 16 sensor connections.

[SYS-HLR-04] Each multiplexer will have a direct or indirect connection path to a singular microcontroller.

[SYS-HLR-05] The microcontroller shall connect to a laptop.

[SYS-HLR-06] When the microcontroller is connected to the laptop, the software shall have access to each sensor's data.

3.3 Software Interfaces

[SYS-HLR-07] Must be using either Windows, Mac, or Linux operating system.

4 Functional Requirements

4.1 Model File I/O

4.1.1 Sample Input/Output

Sample input for this feature would be the user selecting a button which indicates creating a new model. Output for this sample would be that a new empty model is created on the screen. Another sample input is the user clicks a save button for an open model. Output would be the application writing to a file that stores the objective structure of the tiles in relation to the model.

Incorrect sample input and output sequence would be that the user selects creation of a tile and nothing happens.

4.1.2 Associated requirements

[APP-HLR-01] The application shall allow the user to create a model.

[APP-HLR-02] When the user creates a model, the application shall prompt the user the dimensions of each tile will be.

[APP-HLR-03] The application shall allow the user to change the dimensions of all the tiles for a model in settings for that model.

[APP-HLR-04] The application shall allow the user to open any number of models from a file.

[APP-HLR-05] The application shall allow the user to save any open model to a file.

[APP-HLR-06] The application shall allow the user to close any open model.

[APP-HLR-07] When the user closes an open model and the model is modified, the application shall prompt the user to save the model to a file.

[APP-HLR-08] When the user exits the application and there is any open model that is modified, the application shall prompt the user for each modified model to save the model to a file.

4.2 Model Manipulation

4.2.1 Sample Input/Output

Sample input for this feature would be the user clicking on the screen of an empty model. Output would be a prompt for where the user would like to place the first tile (specify position). Another sample input going off the first one is that the user selects a face of one of the tiles. Output would be that the tile is highlighted. Going further, sample input is that the user clicks on a highlighted face, output is that the application generates a tile in that direction.

Incorrect sample input and output would be the user has clicked on a highlighted face and pressed enter, and the application does not exit out of modification mode of a tile.

4.2.2 Associated requirements

[APP-HLR-09] The application shall allow the user to add tiles to an inactive model.

[APP-HLR-10] The application shall allow the user to remove tiles from an inactive model.

[APP-HLR-11] The application shall allow the user to rotate a model.

[APP-HLR-12] The application shall allow the user to translate a model.

[APP-HLR-13] The application shall allow the user to rotate the camera view of a model.

[APP-HLR-14] The application shall allow the user to translate the camera view of a model.

[APP-HLR-15] The application shall allow the user to switch between physical and simulation models for each respective model type.

4.3 Simulation Models

4.3.1 Sample Input/Output

Sample input and output for this feature would be the user clicks on a button to specify an unspecified model to be a simulation model, and the application updates the models specification. Another sample input and output would be the user creates a projectile event before activating a simulation model and activates the simulation model, and the output would be that the simulation has a projectile being fired at the specified time that the user defined in the event.

Incorrect sample input and output would be that a projectile hits the simulation model via a simulation event and the application does not alert the user.

4.3.2 Associated requirements

[APP-HLR-16] The application shall allow the user to specify models to be simulation models.

[APP-HLR-17] The application shall allow the user to specify projectile events for simulation models before activating the simulation model.

[APP-HLR-18] A projectile event shall consist of density, velocity, and starting position of the projectile, as well as the time that the event should be executed within an active simulation model.

[APP-HLR-19] The application shall allow the user to activate a simulation model.

[APP-HLR-20] When a simulation model is active, the application shall execute the projectile events as a function of time.

[APP-HLR-21] The application shall allow the user to create a projectile event during an active simulation model.

[APP-HLR-22] When the user selects a tile in an active simulation model, the application shall indicate the status of the tile by displaying a percentage of health determined by the impact energy of previous impacts on the tile.

[APP-HLR-23] The application shall allow the user to display a percentage of health of all the tiles at once while a simulation model is active.

[APP-HLR-24] When a projectile fired from a projectile event in an active simulation model hits a tile, the impact energy of the projectile on the tile at that specific time will be recorded as data.

[APP-HLR-25] When a simulation model is active, the application shall alert the user if a hit or an internal/external rupture occurs in any of the tiles.

[APP-HLR-26] When a user selects a tile in an active simulation model, the application shall display all data acquired over time for the tile.

[APP-HLR-27] When the application alerts the user of an internal/external rupture of a tile while a simulation model is active, the application shall allow the user to pan to the impact location of the tile.

4.4 Physical Models

4.4.1 Sample Input/Output

Sample input and output of this feature would be that the user clicks on a tile to add connection and the application displays a list of available sensors ordered by their unique identifier.

Incorrect sample input and output would be that the user tries to save connection configuration of the tiles and the application doesn't save it.

4.4.2 Associated requirements

[APP-HLR-28] The application shall allow the user to specify models to be physical models.

[APP-HLR-29] The application shall have a maximum of one open model to be physical.

[APP-HLR-30] The application shall allow the user to add connections from the sensors (connected via the hardware system) for a tile in a physical model.

[APP-HLR-31] The application shall allow the user to edit connections from the sensors for a tile in a physical model.

[APP-HLR-32] The application shall allow the user to remove sensor connections for a tile in a physical model.

[APP-HLR-33] The application shall allow the user to save the connection configurations of a physical model to a file.

[APP-HLR-34] The application shall allow the user to load a connection configuration to a physical model from a file.

[APP-HLR-35] The application shall allow the user to activate a physical model.

[APP-HLR-36] When the user selects a tile in an active physical model, the application shall indicate the status of the tile by displaying a percentage of health determined by the impact energy of previous impacts on the tile.

[APP-HLR-37] The application shall allow the user to display a percentage of health of all the tiles at once while a physical model is active.

[APP-HLR-38] When a physical model is active and a sensor indicates that a projectile has hit a tile, the impact energy of the projectile on the tile (determined by the sensor) at that specific time will be recorded as data.

[APP-HLR-39] When a physical model is active and a sensor indicates that a projectile has internally or externally ruptured a tile, the application shall alert the user of the fact.

[APP-HLR-40] When the application alerts the user of an internal/external rupture of a tile while a physical model is active, the application shall allow the user to pan to the impact location of the tile.

[APP-HLR-41] When the user selects a tile in an active physical model, the application shall display all data acquired over time for the tile.