

Group 10 - Deliverable #2

SFWRENG 3A04: Software Design III - Large System Design

Andrew Hum 400138826

Arkin Modi 400142497

Hongzhao Tan 400136957

Christopher Vishnu 400129743

Shengchen Zhou 400050783

March 5, 2020

1 Introduction

1.1 Purpose

The purpose of the document is to focus on the architecture of the HackerSim system. The system's architecture is based upon business events developed in Deliverable 1 to outline the components of the HackerSim software for both the client and the developer. It covers the architectural decisions that have been made regarding the system and its components. This document is intended for the project manager, the current project team and any future development teams for the HackerSim Project.

1.2 System Description

The HackerSim system is an interactive game that will allow the user to raise a Software Engineer in their room. The main component of our software would be the General Room which is the link that interacts with the rest of the sub-components. The main sub-components that the General Room interacts with which would be the Shop, Friends and Chat, Project and the Time-step. The Shop component focuses on interacting with the inventory for purchasing and browsing items. The Friends and Chat component focuses on providing message functionality between the user and the friends. The Project component focuses on the project and future projects the Software Engineer has to do to gain in-game currency. Finally, the Time-step component focuses on the passage of time and which affects the Software Engineer's attributes.

1.3 Overview

This document is organized by the following sections: Analysis Class Diagram, Architectural Design, Class Responsibility Collaboration Cards. Analysis Class Diagram focuses on providing details about the structure of the classes and their relationships. Architectural Design focuses on the overall architectural design of the HackerSim application, showing the division of the system into subsystems. Finally, Class Responsibility Collaboration (CRC) Cards focus on each individual class and its responsibilities and relations in which they collaborate with other classes.

1.4 Definitions, Acronyms, Abbreviations

SE - Software Engineer

2 Analysis Class Diagram

This section should provide an analysis class diagram for your application.

3 Architectural Design

This section should provide an overview of the overall architectural design of your application. Your overall architecture should show the division of the system into subsystems with high cohesion and low coupling.

3.1 System Architecture

- a) Identify and explain the overall architecture of your system
- b) Be sure to clearly state the name of the architecture
- c) Provide the reasoning and justification of the choice
- d) Provide a structural architecture diagram showing the relationship among the subsystems (if appropriate)

3.2 Subsystems

- a) Provide a brief description of each subsystem. Be sure to document its purpose and relationship to other subsystems.

4 Class Responsibility Collaboration (CRC) Cards

This section should contain all of your CRC cards.

- a) Provide a CRC Card for each identified class
- b) Please use the format outlined in tutorial, i.e.,

Class Name:	
Responsibility:	Collaborators:

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

IMPORTANT NOTES

- Please document any non-standard notations that you may have used
 - *Rule of Thumb*: if you feel there is any doubt surrounding the meaning of your notations, document them
- Some diagrams may be difficult to fit into one page
 - It is OK if the text is small but please ensure that it is readable when printed
 - If you need to break a diagram onto multiple pages, please adopt a system of doing so and thoroughly explain how it can be reconnected from one page to the next; if you are unsure about this, please ask about it
- Please submit the latest version of Deliverable 1 with Deliverable 2
 - It does not have to be a freshly printed version; the latest marked version is OK
- If you do NOT have a Division of Labour sheet, your deliverable will NOT be marked