Deliverable #2 Group #1 T02

SE 3A04: Software Design II – Large System Design

1 Introduction

1.1 Purpose

The purpose of the document is to provide a high level design for the system architecture and class composition. This document is intended for use by the stakeholders of the project as reference throughout the various stages of the SDLC. The primary stakeholders include Prof. Ridha Khedri, who has commissioned this project as part of Software Engineering course SE3A04 as provided by McMaster University, Tutorial Assistants for SE3A04, who are responsible with oversight and review of the deliverables of the project and members of the development team (ie. authors of the document). Other stakeholders include any future teams of developers and managers, that may maintain the application or use this project application as an open source reference.

1.2 System Description

Climatar is a world simulation software system illustrating the long term and short term effects of actions with respect to climate, greenhouse gas levels, economic stability and social relations with the governing bodies in the model world. The model world is based in the Avatar: The Last Airbender universe. Users are given governance of one of the four tribes in the Avatar universe: Air, Water, Fire, and Earth. Based on the state of the world and the regions that compose it, news events will be posed to the user that require action. Depending on their decisions the world will morph around the change and consequences will propagate throughout the simulation. All decisions have consequences associated with them, and the user can only react to events pertaining to their region, removing the omnipotent aspect of control, users must react to the changing world dynamically.

1.3 Overview

The remainder of the document outlines the high level system design specifically outlining how the system will be structured by outlining the Use Case Diagram, Analysis Class Diagram, the inherent Architecture Design, and Class Responsibilities. The document first outlines the use cases for Climatar. The use cases illustrate the business events pertaining to the system and layout the system functionality. Descriptions of each use case aide in formalizing the system functionality from the view points identified previously in the SRS. From the Use Cases a Noun-Verb Analysis was completed to create an Analysis Class Diagram. The Analysis Class Diagram acts as a starting point for the major classes the system will be composed of. The following section, Architecture Design, utilizes the information gained from the analysis class diagram to develop a system architecture to layout the system is a way such that each sub system has high cohesion and low coupling. Finally a Class Responsibility Collaboration (CRC) Card is developed for each class to formally outline the responsibilities, and dependencies for a given class.

2 Use Case Diagram

This section should provide a use case diagram for your application.

a) Each use case appearing in the diagram should be accompanied by a text description.

3 Analysis Class Diagram

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

4 Architectural Design

4.1 System Architecture

The system is divided into a GUI, World and sub system. The World subsystem further divides into Weather, Green House Gases (GHG), Political, and News Event subsystems. The GUI and World sub systems connect to one another but their controls run asynchronously to one another. The selected software architecture for the development of Climatar is Presentation-Abstraction-Control, PAC.

PAC is optimal for the software architecture for Climatar for the aspects as it helps abstract the subsystems and allow for an ease of concurrency between the subsystems, as well as ensuring all modules are loosely coupled such that changes do not propagate through the system. This forces the system to be in a state of being easily maintainable post development. Furthermore, the project has a development time line of approximately one week for the coding portion. PAC allows developers to work in contained subsystems independent of other developers, thus the parallelism for development is optimal.

Provide a structural architecture diagram showing the relationship among the subsystems (if appropriate)

4.2 Subsystems

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

5 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.

Contributor Name	**Contributions**
Wenbin Yuan	
Haris Khan	
Riley McGee	Original draft of 1.* and 4.*
Vishesh Gulatee	
James Taylor	

By signing below you agree to the work divisions stated above correctly representing all contributions made:

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