

# Deliverable #1 T02 Group 2

SE 3A04: Software Design II – Large System Design

February 13, 2017

## 1 Introduction

### 1.1 Purpose

The purpose of the document is to provide a detailed description of the requirements of the application "Climatar". It serves to highlight objectives and key features, the application is perceived to embody, while also describing the scope, constraints and interface of the application. This document is intended for the members of the development team (ie. authors of the document) as reference throughout the various stages of the SDLC.

### 1.2 Scope

Climatar is a UI based mobile game, designed for Android OS, that simulates climate change as experienced by the world made popular by animated TV series, Avatar: The Last Airbender. The game generates news events, which describe real world actions, policies and decisions, which either contribute to or deter climate change, influencing variables that monitor the environmental sphere of the world. These news events are influenced by factors such as energy consumption, political stance, lifestyle and economic motivation and the enactment of these events are determined by the user.

[?]The geography of the model world is divided into four independent nations: Air Nomads, Water Tribe, Fire Nation and Earth Kingdom. The user has the option to choose one of these nations, enacting the news events pertaining to chosen nation or to play god, enacting the news events pertaining to all four nations. Each nation follows a unique political system, culture and lifestyle, and uses the element specified in its name as its primary source of energy. These elements are utilized by special portion of the population known as benders, who have the ability to naturally manipulate a specific element. The game ends once the model world has experienced sufficient damage due to climate change, severely changing the world's eco-system.

The application shall provide users with a save and load game option, allowing the user to play the game, as when they please. The game should be available for download through an application store or public repository.

### 1.3 Definitions, Acronyms, and Abbreviations

1. **BBC**: British Broadcasting Services, a public service broadcaster based in the UK.
2. **Climate Change**: Extended and permanent change, as experienced by long term weather patterns.
3. **GHG**: Greenhouse Gases, classification for gases in the atmosphere that absorb and emit radiation within the thermal infrared range. Major contributor of Global Warming.
4. **Global Warming**: Gradual increase in average surface temperature of Earth.
5. **FPS**: Frames per second, frequency at which a series of images (frames) are displayed, consecutively.
6. **SDLC**: Software Development Life Cycle, described as the framework used to define the hierarchy of tasks to be performed at each stage of the software development process.

7. **SL4A**: Scripting Layer for Android, library used for creating and executing various scripting languages, based in Android systems.
8. **TTS**: Text to speech.

## 1.4 References

## 1.5 Overview

The rest of the document is divided into three sections. The second section provides an abstract overview of the product and its major functionality. The section defines the constraints for the system and the assumptions that affect the systems requirements, all the while highlighting the proposed system's benefits that distinguish it from similar products available. The third section provides a detailed description of the functional requirements perceived, which describe the specific behavior of the system. The contents of this section are classified as business events and viewpoints. The fourth and final section provides a detailed description of the non functional requirements, which describe specific areas of operation of the system.

# 2 Overall Description

This section outlines all details prefacing the requirements for the Climatar project. These details are to be understood prior to the functional and non-functional requirements for the system.

## 2.1 Product Perspective

Climate and world simulation as a software solution is not a new idea, however Climatar as a solution to world events is distinct from the current off-the-shelf solutions, appealing to a new client base, as well as utilizing current solution features to provide an enhanced software system for those using existing off-the-shelf solutions.

Existing products related to Climatar include:

- **BBC - Climate Challenge**

[?]Is a web application for simulating the effects on controlling Great Britain. The application advances with 10 year turns, each turn allows a specific set of decisions to be made. Each decision effects just Great Britain. The overall goal is to become a no-emissions country within 200 years. Climatar contrasts this application through the following points:

- Climatar is for use on android devices. Android web players do not support the application BBC - Climate Challenge.
- Climatar allows different political regions to influence the world attributes, out of the control of the user.
- The over arching goal is to show feasible world reactions to events, elimination of C02 emissions thus is not the only simulation feature.
- Each play through of Climate Challenge is identical, at each phase of the simulation the same options are available. Climatar is to engage users interaction and knowledge growth by ensuring different instances of the simulation will play through differently based on previous actions, illustrating long term effects of decisions.

- **Climate Models of Earth**

Software illustrations of Climate predictions such as the Simple Climate Model by Monash, utilize mathematical predictions through data extrapolation on climate change to illustrate predictions on future climate if all growth rates remain constant with population, C02 levels, average temperature, etc[?].

Climatar contrasts this application through the following points:

- Climatar simulations do not assume constant growth rates, instead all system components like climate simulate and react with on another, causing complex predictions to events to be explored.
  - Applications that model the earth are designed for research use. Climatar is to educate the general public on the effects actions have both long and short term with respect to complex systems which could include, but is not limited to: environment, political toil, sea level, etc.
  - The simulations executed in Climatar will however try and be as accurate to real physical systems, allowing for user analysis of reactions to events in a similar to real world way.
- **Energy Wars** Is a Android game[?] where users control the worlds demand for energy through development of green, and pollutant energy generation schemas. Event responses include natural disasters and war. The game itself is designed to be a puzzle to solve within each provided stage. Climatar contrasts this application through the following points:

- Climatar is to be a simulation solution of the complex systems interacting and handling aspects of regional events and how they effect other regions, whereas Energy wars is designed as a game to and the simulation of the world in the game is non realistic, such that the game moves at a fast pace.
- Both systems utilize the Android platform.
- The user of Climatar is not omnipotent, they may only react to the changing system through events, whereas in Energy Wars users control energy production of every single region, advancements in one relate to advancements in another.
- As opposed to the game modelling the earth, Climatar Models the world present in the Avatar universe. Changing the setting removes the restrictions to the real world, allowing for simulations of generated regions to occur not as the world is, forcing adaptability to the game play aspects of Climatar.

Although many more solutions exist, the above outline all of the main elements to current systems which contrast to Climatar's place in the family of world simulation software solutions.

Climatar is a self-contained software solution with respect to the Android system it is running on. No external databases, servers, or systems are required for Climatar to work. The System itself contains layers for world containment, world components (i.e. Climate, Greenhouse gas levels and emissions, political stances etc.), and event interaction. Depending on the simulated world, each of the sub layers interact differently. Furthermore the System interacts with the android platform utilizing the features available to applications on android devices. The following diagram illustrates the interaction of layers within the system:

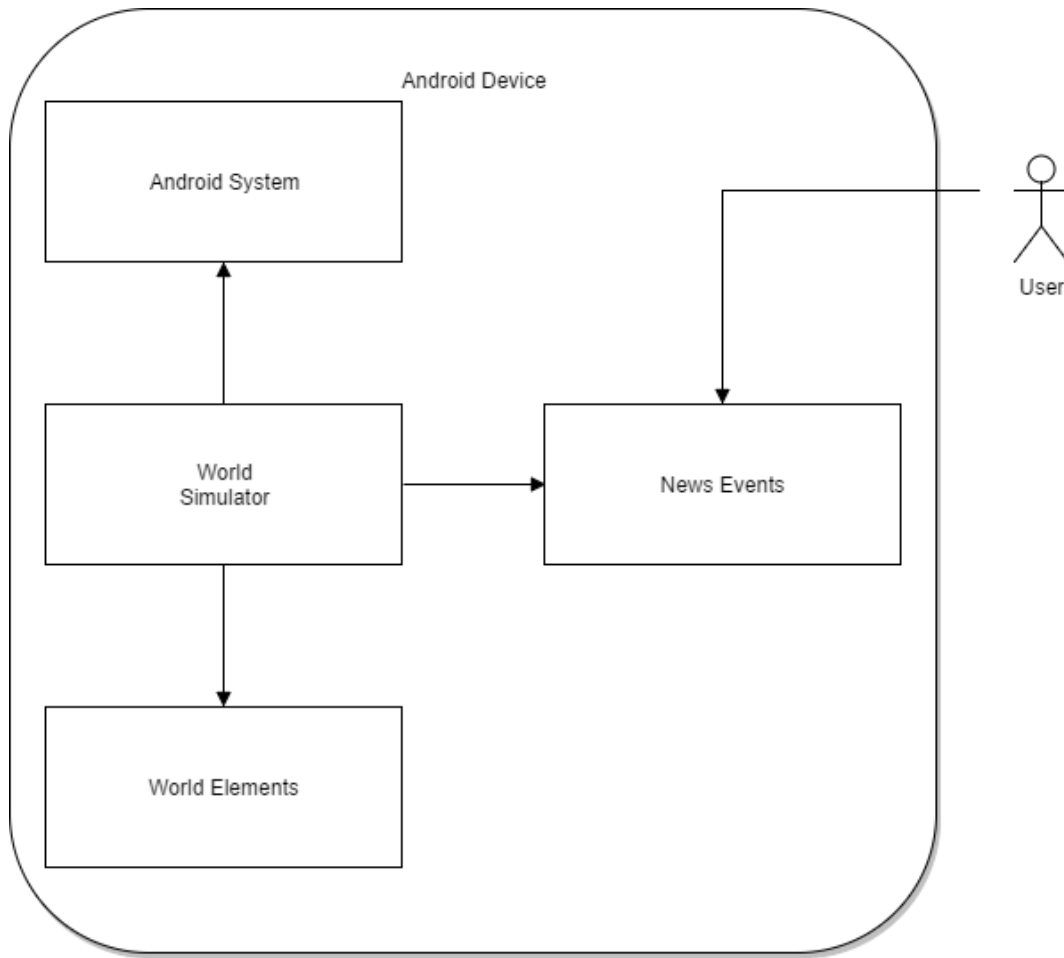


Figure 1: High Level System Interaction Layers.

## 2.2 Product Functions

The functionality Climatar is to present as a product is:

1. Climatar is to present the ability to simulate regions of a world based on regional, political, general climate, and pollution levels of the world under simulation.
2. As a part of simulating the world, news items are presented as attributes of the world, it is up to the user to react to news.
3. User interactions occur with an emphasis on not being omnipotent in how the world works, only in reactions to events within the world itself. In short users are given the ability to react to the changing world, propagating further changes, however they are not given the ability to command all aspects of the world.
4. The application presents the user information allowing them to make needed decisions.
5. Climatar allows for regional analysis, showing specific attributes of regions allowing for contrast in benefits versus disadvantages of regional reactions to changes made.
6. Climatar is a tool to have users think about reactions to events on a global scale in a non real-time scale to show long term effects to event reactions unfold quickly.

## 2.3 User Characteristics

The general characteristics of a user of Climatar are as follows:

- Able to read and comprehend English.
- Understands how to navigate and use an Android system, as well as the controls provided by an Android system.
- The ability to see the system; including:
  - Ability to view objects, and distinct elements on the device.
  - Differentiate colouration of components such as red from blue.

## 2.4 Constraints

1. Climatar is to be restricted to android platforms.
2. Programming language(s) used must be supported by SL4A
3. Climatar must implement a system that is composed of at least three sub-systems, each dealing with diverse environments from which the stimuli are received.
4. Each sub-system must be modularized and easily swappable.
5. Climatar must be able to customize the views to a specific set of sub-systems on the fly.
6. Climatar must react to specific stimuli in a way that respects the modelled domain. Each subsystem must have at least one possible stimuli.

## 2.5 Assumptions and Dependencies

The following list of assumptions hold for the entirety of the SRS, any changes to the below list must be reflective throughout the SRS for it to be considered stable.

- The use of the system assumes that the hardware utilizing the software is running at least Android KitKat 4.4.
- Use of the Java Development Kit 1.8.0 features are available to the system.
- Hardware under use is assumed to have a resolution of at least 720p.
- Climatar is to be granted any system accessibility requests by the user for the system to respond as stated to all requirements.
- Battery life of devices is assumed to remain consistent, the program assumes that the phone does not enter a power down state suddenly.
- Climatar is not planned for release on other platforms apart from Android KitKat 4.4+. Thus all requirements for the system do not include any need for porting the application beyond newer Android versions.

## 2.6 Apportioning of Requirements

Requirements being delayed, thus not included within the SRS are as follows:

1. Online capabilities of the system are not included within the requirements however are alluded to as possible features.

### 3 Functional Requirements

#### BE1. Start Application

##### VP1.1 User Viewpoint

- i. N/A

##### VP1.2 Hardware Viewpoint

- i. The underlying hardware shall create an graphics context. If the device is unable to create a graphics context, the application should close gracefully with an appropriate error message.
- ii. The underlying hardware shall be initialized to respond to any touch input from the user.
- iii. The hardware shall be able to load files that contain user data and save data, If the hardware fail to load the required files, the application should ask user to create a new game.
- iv. The hardware should be able to render the start menu and refresh at realtime (24 FPS).

##### VP1.3 Game World Viewpoint

- i. N/A

#### BE2. Terminate Application

##### VP2.1 User Viewpoint

- i. The user should be capable of quitting the game at any time using the back button.
- ii. The user should have the option of saving a game at any time.
- iii. The user should be capable of enabling an autosaving option.

##### VP2.2 Hardware Viewpoint

- i. The hardware shall dispose of the graphics context.
- ii. Touch events received from the hardware shall be ignored.
- iii. If the autosaving option is enabled, the hardware will save the game files if capable, else display an appropriate error notification.

##### VP2.3 Game World Viewpoint

- i. N/A

#### BE3. Pause Game

##### VP3.1 User Viewpoint

- i. The user should be capable of saving the game while it is paused.

##### VP3.2 Hardware Viewpoint

- i. The graphics context shall be updated to display the pause screen.
- ii. If the back button event is captured by the hardware, the game shall resume.

##### VP3.3 Game World Viewpoint

- i. The system shall stop generating game events.
- ii. The system shall stop the game timer.

#### BE4. Resume Game

##### VP4.1 User Viewpoint

- i. N/A

##### VP4.2 Hardware Viewpoint

- i. N/A

##### VP4.3 Game World Viewpoint

- i. The System shall start generating events.
- ii. The System shall resume the timer.

#### BE5. Start Game

##### VP5.1 User Viewpoint

- i. The user should be able load a previous saved game.
- ii. The user should be able to start a new game by selecting a game mode.
- iii. There shall be two game modes:
  - A. Survival mode
  - B. Overlord mode
- iv. The user should be allowed to respond to any game events by selecting ?YES? or ?NO? in the UI.

##### VP5.2 Hardware Viewpoint

- i. The hardware shall load the game file which is selected by the user.
- ii. The hardware shall load any graphics assets which are required for rendering the game world.
- iii. While the assets are loading, a loading screen shall be presented to the user.
- iv. After all assets are loaded, the game world and any UI shall be rendered in the graphics context.
- v. The hardware shall update the graphics context to display any new game events that occur.

##### VP5.3 Game World Viewpoint

- i. The game world shall initialize data when a new game is started.
- ii. If the game world is loaded, the game world state shall reflect that of the loaded game file.
- iii. The game world shall begin the game timer.
- iv. The game world shall simulate and begin generating game events once it is started or loaded, based on the current game world state.
- v. Game events shall be yes or no questions which result in an increase or decrease of the chaos, temperature and or money.

#### BE6. Move map

##### VP6.1 User Viewpoint

- i. The user should be able to move the map with the finger.

##### VP6.2 Hardware Viewpoint

- i. The hardware should be capable to accept user?s finger input.
- ii. The graphics context shall be updated to display the movement of map with realtime (24FPS).

##### VP6.3 Game World Viewpoint

- i. N/A

#### BE7. Input of decision

##### VP7.1 User Viewpoint

- i. The user shall respond to game events by selecting ?YES? or ?NO?.

##### VP7.2 Hardware Viewpoint

- i. The hardware shall update the chaos, politics and money values on screen.

##### VP7.3 Game World Viewpoint

- i. The game world shall appropriately increase or decreases the chaos, politics, and money values based on the decision of the user.

- ii. The game world shall check for lose conditions by checking if the following is truth: (chaos  $\geq$  100 percent OR money  $\leq$  0 OR temp  $\geq$  100 percent).
- iii. If the lose conditions are truthy, the game shall display the gameover screen.

#### BE8. Reset Game

##### VP9.1 User Viewpoint

- i. The user shall choose yes or now to confirm resetting the game world.

##### VP9.2 Hardware Viewpoint

- i. The hardware shall render and update the display with new map and initialized UI.

##### VP9.3 Game World Viewpoint

- i. The System shall initialize all data and generate new map.
- ii. The System shall start generating game events.

#### BE9. The game ends

##### VP9.1 User Viewpoint

- i. N/A

##### VP9.2 Hardware Viewpoint

- i. The graphics context shall be updated to display the game over screen with score, highest score, restart button and quit button.

##### VP9.3 Game World Viewpoint

- i. The system shall be terminated and all temporary data shall be cleaned.

## 4 Non-Functional Requirements

### 4.1 Look and Feel Requirements

#### 4.1.1 Appearance Requirements

- LF1. The product's interface shall be designed in a way such that it can be easily adapted to different screen sizes.
- LF2. The product shall minimize the use of visual adornments that would otherwise clutter the interface.
- LF3. The product shall use common navigational elements (ie. status bar, tab bar, hamburger menu) to help usability amongst new users.
- LF4. The product shall use common iconography to help usability amongst new users.
- LF5. The product shall make heavy use of contrast to help it's usability amongst colour blind users.
- LF6. The product shall predominantly use a single colour to help develop and maintain it's brand identity.

#### 4.1.2 Style Requirements

- LF7. Where it can, the product shall conform to Google's material design guidelines.
- LF8. The product shall use primarily sans-serif fonts.
- LF9. The product's app icon shall be designed in the common negative-space style to help it sit better amongst other popular app icons.



## **4.2 Usability and Humanity Requirements**

### **4.2.1 Ease of Use Requirements**

- UH1. The product shall be usable by anyone fluent in the English language.
- UH2. The product's navigational elements should be found in under 2 seconds by new users.
- UH3. The product's navigational elements shall behave in the ways standard to the Android operating system. This should make the app more intuitive to use for new users.
- UH4. The product's functions shall be designed a way such that they require a minimum number of gestures from the user.
- UH5. Where it makes sense, the product shall favour input elements such as sliders over raw text input.

### **4.2.2 Personalization and Internationalization Requirements**

- UH6. The product shall favour the use of icons such that, when needed, it can be more easily adapted to different languages and cultures.
- UH7. The product shall use Android SDK best practices so that the user's global accessibility settings (text size, text-to-speech etc...) will take effect in the app.

### **4.2.3 Learning Requirements**

- UH8. By using standard Android navigational elements, the product should not take more than 10 minutes for a new user to become adept with it.
- UH9. For any non-traditional feature, the product shall use short tutorial text to educate the user about it's functionality.

### **4.2.4 Understandability and Politeness Requirements**

- UH10. The product shall contain no language or ideas inappropriate for an elementary school level user.

### **4.2.5 Accessibility Requirements**

- UH11. The product's interface shall use intractable elements at a minimum of 60x60points in size to help users with reduced motor skills.
- UH12. By employing standard Android SDK best practices as mentioned in LF7, the product shall adapt the user's global accessibility settings such as TTS (text to speech) and increased/decreased font sizes.
- UH13. The product shall make heavy use of contrast and minimize the use of multiple colours to help colour blind users (This is also mentioned in LF5).

## **4.3 Performance Requirements**

### **4.3.1 Speed and Latency Requirements**

- PR1. Any network requests made by the product will be dispatched and listened to on a background thread so that the user interface (driven by the main thread) does not stall.
- PR2. Each user interaction shall be followed up by a visual change on screen. The time between these two events should never exceed half a second.
- PR3. Procedural world generation for a new game should take no longer than 2 seconds.
- PR4. Loading an in-progress game should take no longer than 1 second.

- PR5. The app should launch in no longer than 2 seconds.
- PR6. The app should resume from background in no longer than 1 second.

#### **4.3.2 Safety-Critical Requirements**

n.a

#### **4.3.3 Precision or Accuracy Requirements**

- PR7. When performing floating-point arithmetic, the app shall favour the use double precision to minimize rounding error.

#### **4.3.4 Reliability and Availability Requirements**

- PR8. The product shall not require an active internet connection (ie. it will be available offline).
- PR9. In any case that the device is running under normal circumstances, the app shall be available.
- PR10. Under any normal use case, the app should not crash.

#### **4.3.5 Robustness or Fault-Tolerance Requirements**

- PR11. The app shall temporarily save an ongoing game in the event that it is terminated unexpectedly by the user or by the operating system in the event that memory needs to be freed.
- PR12. The app shall reopen the saved game from PR11 in the event that it is launched following an unexpected termination.

#### **4.3.6 Capacity Requirements**

- PR13. The product shall allow the user to save up to 10 in-progress games. This does not include the game temporarily saved in the event that the app terminates unexpectedly (PR11).

#### **4.3.7 Scalability or Extensibility Requirements**

- PR14. The product is planned to feature an online versus mode in the future in which it will allow for up to 4 concurrent players per session.

#### **4.3.8 Longevity Requirements**

- PR15. The product should be maintainable and extensible within it's planned maintenance budget for 5 years post launch.

### **4.4 Operational and Environmental Requirements**

#### **4.4.1 Expected Physical Environment**

- OE1. As the app operates within the Android environment and introduces no restriction under which it operates within that environment, the product shall be usable under circumstances where the Android operating system allows it.

#### **4.4.2 Requirements for Interfacing with Adjacent Systems**

- OE2. The product shall work on any smartphone running Android 4.4 (KitKat) and later.
- OE3. As an online feature is planned for sometime in the future, the app shall take into consideration architectures for interfacing with a socket based web server.
- OE4. Future updates to the app shall not invalidate game saves made on previous versions.

#### **4.4.3 Productization Requirements**

- OE5. The product shall be installable directly from the Google Play Store.
- OE6. The product shall have 5 screenshots and a short text description to advertise it on the Google Play Store.
- OE7. The product shall be no bigger than 15MB.

#### **4.4.4 Release Requirements**

- OE8. Maintenance and new content updates will be published at the end of each month.

### **4.5 Maintainability and Support Requirements**

#### **4.5.1 Maintenance Requirements**

- MS1. Hotfixes must available to publish within a day of them being made.

#### **4.5.2 Supportability Requirements**

- MS2. A link to the company's support url will be provided in the description on the Google Play Store for feature requests and bug reports.

#### **4.5.3 Adaptability Requirements**

- MS3. The product is expected to run on version of Android 4.4 (KitKat) and later.
- MS4. The product may be ported to iOS and Windows Phone in the future.
- MS5. The product may be adapted to accommodate different screen sizes and aspect ratios in the future.

### **4.6 Security Requirements**

#### **4.6.1 Access Requirements**

- SR1. When online features are introduced in the future, only a user logged into an account may play under that account's identity.

#### **4.6.2 Integrity Requirements**

- SR2. The product shall guard against the introduction of incorrect data to, for example, guard against an attacker spoofing high scores numbers.
- SR3. When online features are introduced in the future, the product shall guard against the automated generation of user accounts.

#### **4.6.3 Privacy Requirements**

- SR4. The product shall inform users of its information policy before collecting data on them.
- SR5. The product shall inform the users of changes to its information policy.
- SR6. The product shall adhere to it's information policy if revealing private user information.

#### **4.6.4 Audit Requirements**

n.a

#### **4.6.5 Immunity Requirements**

- SR7. As the product runs under management of the Android operating system, the immunity of the product relies on the guards put in place by Android.

### **4.7 Cultural and Political Requirements**

#### **4.7.1 Cultural Requirements**

- CP1. The product shall aim to not offend any ethnic or religious groups.
- CP2. The product shall hold seasonal events celebrating holidays from a variety of ethnic and religious backgrounds.

#### **4.7.2 Political Requirements**

n.a

### **4.8 Legal Requirements**

#### **4.8.1 Compliance Requirements**

- LR1. Personal information shall be implemented in compliance with the data protection act.

#### **4.8.2 Standards Requirements**

- LR2. The app will conform with acceptable performance standards for games and simulations.
- LR3. The app will conform to the Android Core App Quality standards to minimize turnover approval time and to increase the chances of being featured in the Google Play Store.

## A Division of Labour

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

Contributor Name	Contributions
Leo Yuan	Section 3.* Functional Requirements Planning and Formalization in association with Haris
Haris Khan	Section 3.* Functional Requirements Planning and Formalization in association with Leo
Riley McGee	Section 2.* Overall Description
Vishesh Gulatee	Section 1.* Introduction
James Taylor	Section 4.* Non-Functional Requirements

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

## IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
  - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or underline
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.