

# Curtin University – Department of Computing

## Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:		Student ID:	
Other name(s):			
Unit name:		Unit ID:	
Lecturer / unit coordinator:		Tutor:	
Date of submission:		Which assignment?	(Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature: \_\_\_\_\_ Date of signature: \_\_\_\_\_

*(By submitting this form, you indicate that you agree with all the above text.)*

# Contents Page

<b>Status</b>	<b>2</b>
<b>Highlights</b>	<b>2</b>
<b>Progress Tracking</b>	<b>5</b>
Toggle Report	5
Activities and Evidence	6
Reflection	6

## Status

Ongoing work commitments can make contributing to this unit (as well as my three other units) challenging, on occasion. However, I have still been able to accomplish all of the tasks which have been assigned to me up until this point. Should I begin to fail to meet these commitments, I will communicate with my employer to perhaps reduce my scheduled hours to a more manageable amount.

## Highlights

- Solely contributed to the completion of the following sections of the SAS for drafts 1 and 2; and the final submission:
  - Quality Attributes
    - Reliability
    - Availability
    - Serviceability
    - Best Practice Commentary
    - Architecture Commentary
  - Phasing, Roadmap and Implementation Plan
    - Technical Assumptions
    - Solution Phases and Technical Milestones
    - Implementation Instructions

*View the below screenshots for evidence of my contributions—note, these screenshots represent the state of the SAS as per draft 2, view the final version for more detailed evidence:*

## 7 Quality Attributes

In discussion of the quality attributes applicable to the solution, specific scenarios relating to each attribute—namely, reliability, availability and serviceability—and in turn, the solution, will be discussed with use of *quality attribute scenarios*. Consider a quality attribute scenario as a brief description of how the application is required to respond to some stimulus.

As per (Carnegie Mellon University 2015) there are six parts which comprise a quality attribute scenario. In order, they are:

1. *Source*: some entity (user, developer, the application etc) which generates a stimulus
2. *Stimulus*: a condition which affects the application
3. *Artifact*: the part of the application which was stimulated by the stimulus
4. *Environment*: the condition under which the stimulus occurred
5. *Response*: the activity to be completed as a result of the stimulus
6. *Response measure*: the measure by which the response will be evaluated

## 7.2 Availability

### 7.2.1 Malicious Attack

**Scenario:** *The application is compromised by a Denial of Service (DoS) attack*

**Source:** External to the application

**Stimulus:** Application receives more requests per second than it can process

**Artifact:** External to the application

**Environment:** Normal operation

**Response:** The client blocks concurrent user requests, displaying a message to the user informing them to wait momentarily before attempting that particular task again.

**Response Measure:** The DoS attack does not persist for longer than one minute.

### 7.2.2 Unresponsiveness

**Scenario:** *The Spotify platform becomes unresponsive, causing the application to fail*

**Source:** Internal to the application

**Stimulus:** A fault in the Spotify platform

**Artifact:** The Spotify platform

**Environment:** Normal operation

**Response:** The application handles the resultant failure, and shuts down until the Spotify platform returns to normal operation

**Response Measure:** The uptime of the application is at least 95%

## 9. Phasing, Roadmap, and Implementation Plan

### 9.1 Technical Assumptions

#### 9.1.1 User Interfaces

##### **Mobile and Desktop**

The UI belonging to the application is compatible with both mobile and desktop devices, and additionally, supports a dynamically scalable UI in order to provide a consistent UX between a variety of different mobile and desktop devices.

#### 9.1.2 Hardware Interfaces

##### **Internet Capability**

The User will access the application via an internet-enabled mobile phone, tablet, laptop or desktop computer. In addition, if the User is the Host, they will require access to a speaker (or speakers) connected to their device either internally or externally, to act as an audio output for the application.

#### 9.1.4 Communications Interfaces

##### **Web Browser**

The User will be accessing the Virtual Jukebox application only via either of the latest two publicly released versions of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari.

##### **Client–Server Communication**

In addition, client–server communication will be facilitated by Django—for database integration; HTTPS for encrypted information transfer between, namely, the application, User, and the Spotify Platform; and WebSocket to facilitate two-way communication between the aforementioned entities.

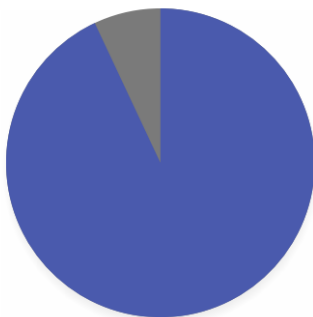
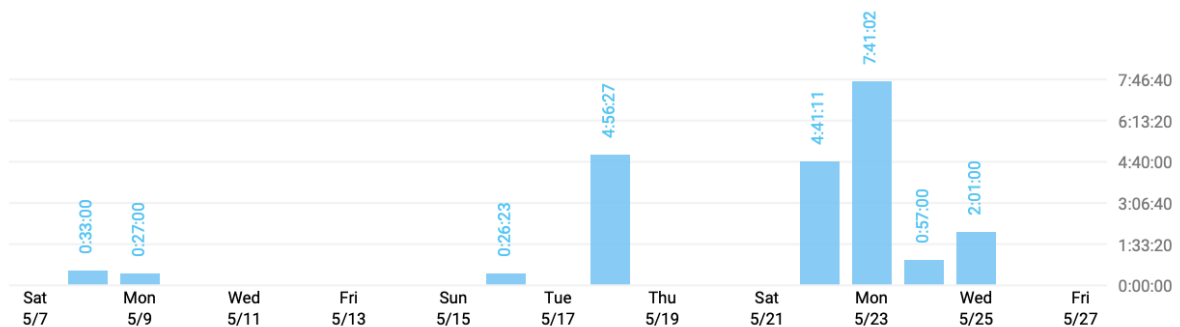
# Progress Tracking

## Toggle Report

### Summary Report

05/07/2022 – 05/27/2022

TOTAL HOURS: 21:43:03

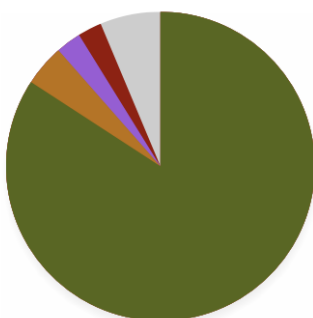


#### PROJECT

- Virtual Jukebox application
- Amristar
- Without project

#### DURATION

20:12:41  
1:30:22



#### TIME ENTRY

- Package 3.1. Solution Architecture Specification
- Sprint 5 Standup Meeting 1
- Sprint 5 Kick-off Meeting
- Sprint 4 Standup Meeting 1
- Other time entries

#### DURATION

18:15:52  
0:57:00  
0:34:23  
0:33:00  
1:22:48

## Activities and Evidence

### **Package 3.1. Solution Architecture Specification**

Represents the work done in order to complete *Package 3.1. Solution Architecture Specification*. The final document is accessible at the following link:

[Software\\_Architecture\\_Specification\\_\(v1.0\)](#)

### **Kick-off Meetings**

My presence in each of these meetings is noted in the minutes documents accessible at the following link: [Kick-off Meetings](#)

### **Retrospective Meetings**

My presence in each of these meetings is noted in the minutes documents accessible at the following link: [Retrospective Meetings](#)

### **Supervisor Meetings**

My presence in each of these meetings is noted in the minutes documents accessible at the following link: [Supervisor Meetings](#)

## Reflection

The Solution Architecture Specification is on-track to be delivered to the client in a reasonable manner. However, heavier workloads due to exams has made it difficult to prioritise capstone. Fortunately, this document will be submitted before exam study is in full effect.