



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

---

## FUNCTIONAL SPECIFICATION

---

PROJECT NAME: ARCANE ARCADE

CLIENT: TONY VD LINDEN

TEAM NAME: TERABITES

### TEAM MEMBERS

NG Maluleke	D Mulugisi	C Nel	LE Tom
13229908	13071603	14029368	13325095

August 21, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Functional Requirements and Application Design</b>	<b>3</b>
2.1	Use Case Prioritization . . . . .	3
2.1.1	Critical . . . . .	3
2.1.2	Important . . . . .	3
2.2	System Modules . . . . .	3
2.2.1	Arcane Arcade High Level Diagram . . . . .	3
2.2.2	Authentication Module . . . . .	4
2.2.3	User Management Module . . . . .	4
2.2.4	Challenge Management Module . . . . .	4
2.2.5	Compiler Module . . . . .	5
2.2.6	Player Interaction Module . . . . .	5
<b>3</b>	<b>Open Issues</b>	<b>6</b>

# 1 Introduction

The project is called *Arcane Arcade*, which references the esoteric language users will have to use, as well as the gamification approach to try and make it as fun as possible. It revolves around building a platform that can be used to gauge the programming aptitude and capabilities of existing or future software developers using a custom esoteric programming language. The platform which is to be available on both mobile and online is thus required to test prospective employees while keeping the tests fun by using gamification principles. The tests should also indicate in which area the user is likely to be better suited, by looking at how the user completed the challenges, such as whether the user used any hints and the badges that the user earned.

## 2 Functional Requirements and Application Design

This section outlines the required functionalities from the system.

### 2.1 Use Case Prioritization

Priorities of the system's use cases.

#### 2.1.1 Critical

- **User Authentication** - This is critical as it limits the system's access only to people who have been authorized. Only registered users are then able to complete the challenges and/or modify the challenges together with authorizing more users. No one can then access any part of the system unless they have been registered and authorized to request such services.
- **Player Interaction** - Interaction is critical as the whole system's core functionality revolves around it. The players have to be able to view and take challenges that are open to them at that time.
- **Code Compilation** - Code compilation is a critical use case as other instances of Player Interaction revolve around it. Where a Player interacts by typing code the system needs to be able to compile that code.

#### 2.1.2 Important

- **Challenge Management** - This is considered important because this is where users who have management privileges will configure the structure of the challenges such as the number of levels and questions. The managers will also be able to set hints and other items related to the challenges.
- **User Management** - This use case is important as it deals with adding and removing users to the system.

### 2.2 System Modules

The system will be built as a series of separate modules. This approach makes it possible to add more modules at a later stage and to modify one module without interfering with the others. It then allows system to be pluggable and integrable.

#### 2.2.1 Arcane Arcade High Level Diagram

**Insert the Arcane Arcade System use case diagram here.**

The core of the Arcane Arcade system. It provides various functionalities to different types of users but the sub-systems offer high level use cases are shown in the diagram.

### 2.2.2 Authentication Module

This module provides the functionality which validate the user credential to check if they match up with the access that they request. The most crucial part and main focus of the authentication module is to verify if a user has access to certain pages given the type of user they are. This then improves the system's security and keeps the data confidential as it controls who has access to which functionality.

**Insert the authenticate module use case diagram here.**

### 2.2.3 User Management Module

This module focuses on the ability to grant and revoke user access to the system. Only the super user has access to the module and it can be used to add, remove and update user accounts. **Insert the Challenge Management module use case diagram here.**

- **addUser** - The addUser use case falls under User as depicted in the above diagram. The manager has the ability to add more user accounts via the user management page thereby allowing them to access the system. The service contract and activity diagram for addUser are shown below.
- **updateUser** - The updateUser use case falls under User as depicted in the above diagram. The manager has the ability to modify user accounts items via the user management page. The service contract and activity diagram for updateUser are shown below.
- **removeUser** - The removeUser use case falls under User as depicted in the above diagram. The manager has the ability to remove any user account via the user management page thereby restricting them from accessing the system. The service contract and activity diagram for removeUser are shown below.

### 2.2.4 Challenge Management Module

Only super users(management accounts) have access to this module.It provides the ability to add, remove and modify the structure of challenges. It allows the manager to modify the challenge at any of its three levels which are challenge, level and question. The modifications that the manager can do are add, remove and update any of the three levels of a challenge.

**Insert the Challenge Management module use case diagram here.**

### **2.2.5 Compiler Module**

### **2.2.6 Player Interaction Module**

This module houses the system's main functionality. The main reason of having the system is to allow the users to interact with the system and complete the offered challenges. This use case diagram indicates the functionality around interacting with the system as a player.

### 3 Open Issues