



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

SOFTWARE REQUIREMENTS 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

May 21, 2016

Contents

1	Introduction	1
2	Vision	1
3	Background	1
4	Architecture Requirements	2
4.1	Access channel requirements	2
4.2	Quality requirements	2
4.3	Integration requirements	2
4.4	Architecture constraints	2
5	Functional requirements and application design	3
5.1	Use case prioritization	3
5.2	Use case/Services contracts	3
5.3	Required functionality	3
5.4	Process specifications	3
5.5	Domain Model	3
6	Open Issues	4

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.

2 Vision

The vision is to provide a fun and easily accessible platform that can be used to gauge the programming aptitude and capabilities of existing or future software developers using a custom esoteric programming language.

3 Background

There are many models in which to ascertain whether a prospective employee has the required skills or aptitude to be a valuable software developer. On-line assessments are easy to execute, but lack the insight provided by manual and proprietary testing of how the candidate reasons. Manual and proprietary testing on the other hand is time-consuming and expensive to execute. In addition, company proprietary tests become stale and are leaked into the industry reducing the value tests may have.

BBD has thus opted to use an esoteric programming language in order to test the skills and aptitude of prospective employees, regardless of their programming experience or preferred programming language. A mobile and online platform 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.

4 Architecture Requirements

4.1 Access channel requirements

The system requires two interfaces:

- Web interface
- Android app (Could also be a mobile website)

4.2 Quality requirements

The following quality aspects needs to be addressed:

- **Scalability:** The system will be used by mimimum of one administrator and the number of players (potential employees) is infinitely.
- **Security:** Passwords will be hashed with the SHA512 hashing algorithm.
- **Reliability:** An automatic partial dump backup can be done everyday at 00:00 and each back up will be kept for a week. On top of this a full dump will be done once a week and will be kept for at most 3 weeks. Alternatively the backup can be done by an external program.
- **Audibility:** Every change should be logged and the person responsible for that log will be recorded. This will allow the administrators to track the users information/data.
- **Maintainability:** Data that is deleted will only be moved to another database that might be a little slower. This will help alleviate some pressure off the main database.
- **Cost:**

4.3 Integration requirements

4.4 Architecture constraints

5 Functional requirements and application design

5.1 Use case prioritization

5.2 Use case/Services contracts

5.3 Required functionality

5.4 Process specifications

5.5 Domain Model

6 Open Issues