System Requirements Specification for: UNTITLED GAME (Project Walrider?)

Hesham H. Salman Jonathon Kissinger Sean Mead Troy Johnson

October 21, 2014

1 Introduction

1.1 Overall Description

UNTITLED GAME tracks user inputs and behavioral patterns, and uses this data to determine whether or not the user is likely to have ADHD. This is determined by comparing the user response times and input patterns to those of people with and without ADHD. If the application can determine with a high level of confidence that the user is likely to have ADHD, a suggestion is made to the user to visit a medical professional.

This application is to be used by a number of people with and without ADHD so that we may test its detection algorithms.

1.2 Definitions

Acronyms and abbreviations as applicable

1.3 References

References to other documents

1.4 Overview of Developer's Responsibilities

Troy Johnson: Jonathon Kissinger: Sean Mead: Hesham Salman:

Responsibilities in terms of development, installation, training, maintenance etc.

Eg. The developer is responsible for development, installation, loading of all the data relating to books in the library, and training of employees.

2 General Description

2.1 Product Perspective

This product will interface with a server, transferring anonymized local user data to a remote server for data processing.

2.2 Product Functions Overview

1. Game: Hurdle Jumper
2. User input tracking and data collection

3. Data processing to determine whether or not the user has ADHD
4. Result generation
5. Comparative analysis of user's data with average user data

2.3 User Characteristics

The users of this application are to be people with and without ADHD. A user survey is conducted at the start of this application such that the results can be calibrated based on the user's age and gender. Of users with ADHD, there is a distinction made between users that are medicated and unmedicated.

2.4 General Constraints

This project is to be completed, including coding, testing, and loading of the database by Nov 30, 2014.

3 Information Description

3.1 Entities and Relationships

Give a list of entities/relationships that are needed, and ER diagrams.

3.2 Data Dictionary

Give the relations, their attributes, and for each attribute the type and a description of the attribute. Here's an example.

1. customers

account_no	varchar(8)	primary key	account number
name	varchar(20)	not null	Name
profession	varchar(10)	not null	Profession
address	varchar(40)	not null	Address
email	varchar(40)	-	Email Address

3.3 Data Flow

Give data flow between major units of your software. E.g., Most useful in case a task has multiple steps requiring interaction with other software or other humans. E.g. To purchase a book, user can enter request, department head can approve, then librarian approves, then order is made.

4 Functional Requirements

Give a description of each process/function. Start from what happens when the user connects to the system. E.g. The main screen and alternatives available to choose from would be a functional requirement. Authentication when entering the system would be another. An interface to issue a book would be a third.

Have one subsection for each functional requirement.

4.1 Functional Requirement 1

Note what a particular functionality does. Make sure to also note what happens in special cases such as out of stock.

4.2 Functional Requirement 2

...

5 External Interface Requirements

5.1 User Interfaces

This used to be important, and required a user manual giving commands screen formats, outputs, error messages, etc. In your case just say the web is used as interface.

5.2 Hardware Interfaces

Interfaces with existing as well as new or special purpose hardware. E.g. smart card reader for authentication. In your case say no special hardware.

5.3 Software Interfaces

List software platforms that are to be supported.

List interfaces with other software packages etc. Make sure to list all software that you interface with.

6 Performance Requirements

Capacity requirements (estimates of no of users, database size, ..) Response time requirements (for different tasks) and throughput (estimates of avg/peak number of transactions per minute, etc). Don't worry about exact numbers but give ballpark figures.

7 Design Constraints

7.1 Standards Compliance

Software development standards asn well as organizational standards (eg reports, auditing).

In your case, say not applicable.

7.2 Hardware Limitations

Requirements on machines, operating systems, storage capacities etc. This used to be very important, but these days this matters only for very fast/highthroughput systems, not so much for the things you are doing.

Fault tolerance and reliability requirements should also be listed here.

In your case this is not considered

8 Validation Criteria

Validation criteria for demonstrating successful implementation and user acceptance. In general includes classes of tests and expected software response.

E.g. I will load the database with X amount of data for testing, and an overview of tests you will run.

9 Other Requirements