Discrete Math Quiz Game with Voice Recognition Requirements Specificatio

Version 1.0

August 28, 2018

Table of Contents

[1. Executive Summary 3](#_Toc191724230)

[1.1 Project Overview 3](#_Toc191724231)

[1.2 Purpose and Scope of this Specification 3](#_Toc191724232)

[2. Product/Service Description 3](#_Toc191724233)

[2.1 Product Context 3](#_Toc191724234)

[2.2 User Characteristics 3](#_Toc191724235)

[2.3 Assumptions 3](#_Toc191724236)

[2.4 Constraints 3](#_Toc191724237)

[2.5 Dependencies 4](#_Toc191724238)

[3. Requirements 4](#_Toc191724239)

[3.1 Functional Requirements 5](#_Toc191724240)

[3.2 User Interface Requirements 5](#_Toc191724241)

[3.3 Usability 5](#_Toc191724242)

[3.4 Performance 6](#_Toc191724243)

[3.4.1 Capacity 6](#_Toc191724244)

[3.4.2 Availability 6](#_Toc191724245)

[3.4.3 Latency 6](#_Toc191724246)

[3.5 Manageability/Maintainability 6](#_Toc191724247)

[3.5.1 Monitoring 6](#_Toc191724248)

[3.5.2 Maintenance 6](#_Toc191724249)

[3.5.3 Operations 6](#_Toc191724250)

[3.6 System Interface/Integration 7](#_Toc191724251)

[3.6.1 Network and Hardware Interfaces 7](#_Toc191724252)

[3.6.2 Systems Interfaces 7](#_Toc191724253)

[3.7 Security 8](#_Toc191724254)

[3.7.1 Protection 8](#_Toc191724255)

[3.7.2 Authorization and Authentication 8](#_Toc191724256)

[3.8 Data Management 8](#_Toc191724257)

[3.9 Standards Compliance 8](#_Toc191724258)

[3.10 Portability 8](#_Toc191724259)

[4. User Scenarios/Use Cases 9](#_Toc191724260)

[5. Deleted or Deferred Requirements 9](#_Toc191724261)

[6. Requirements Confirmation/Stakeholder sign-off 10](#_Toc191724262)

[APPENDIX 11](#_Toc191724263)

[Appendix A. Definitions, Acronyms, and Abbreviations 11](#_Toc191724264)

[Appendix B. References 11](#_Toc191724265)

[Appendix C. Requirements Traceability Matrix 11](#_Toc191724266)

[Appendix D. Organizing the Requirements 13](#_Toc191724267)

# Executive Summary

## Project Overview

The Accounting and Finance is an online integrated financial accounting system that records, monitors and maintains all accounting and financial transactions of the organization. The Accounting and finance system provides a complete financial budgeting and managerial reporting system that, due to its flexibility, is able to meet the financial reporting needs of the organization.

The Accounting and Finance system is a system of collecting, storing and processing financial and accounting data that are used by decision makers. This system is generally a web-based method for tracking accounting activity. The resulting financial reports can be used internally by management or externally by management or externally by other interested parties including investors, creditors and tax authorities. Accounting and Finance are designed to support all accounting functions and activities including auditing, financial accounting and reporting managerial and tax.

## Purpose and Scope of this Specification

The purpose of the system is to collect, store and process financial and accounting data, to accumulate and report on financial information about a firm’s performance, financial position and cash flows of system. To also manage a business’ records to keep track of income, expenses and other financial activities.

***Include a description of what is within the scope what is outside of the scope of these specifications. For example:***

In scope

* Produce calculation for annualization, final pay that are definitive and irrefutable
* Preparing employee’s payroll including calculation of salaries, contribution for social insurance, health insurance, unemployment insurance and personal income tax.
* Preparation of corresponding payment orders and transfer orders for employees salaries
* Preparing and filling monthly and annual salary returns
* Registration/ deregistration of employees with relevant social institution
* Preparation of calculation for other contract per client request

Out of Scope

* Accepting/hiring employees
* Product supplying and Delivery
* Direct collecting of payments from customer
* Managing employees
* Viewing products and it’s price
* Storing of products
* Advertising of products

# Product/Service Description

The general factor of this application and goal of this study is to provide information about how the user can easily understand the subject of discrete mathematics by easily reading some algorithms, discussion, and topics that the user can use to answer the questions; it is a practice point to improve the effectiveness for those users.

## Product Context

The application like this is the same to other product that the user wants to learn. This application is self-contained. This application is creating a good influence the user and user are needed to learn using this application.

## User Characteristics

Create general customer profiles for each type of user who will be using the product. Profiles should include:

* Students
* experience
* know to follow the game instruction

## Assumptions

This is available for the entire user that is interested to the discrete mathematics, showed that providing feedback on progress toward learning goals increases performance and formative assessment provides information to identify gaps between current and desired performance, which a learner may then strive to narrow to maximize performance.

## Constraints

Describe any items that will constrain the design options, including

* No security, it is open to all user :access, management and security
* Connection Interrupt
* The system has 15 second per question.
* While using Android Studio due to lack specs needed of the device that cause logging

## Dependencies

List dependencies that affect the requirements. Examples:

* This application required internet to play the multiplayer.
* This application made by Android Studio.

# Requirements

* The system was made through the android studio, In this process you will know if the program was working or not. Structured Query Language is standard computer language for relational database management and data manipulation
* The system input will come from user’s, the user need to input the answer to each question contained in each level then its output will show the user the list of answer each level, the function of each output will eventually be seen by the last part of level
* The priority of the system is the user that having difficulty on how to understand the discrete math. Nowadays most of the user use website and application that why this application may help those users to understand more on how solve the problem in discrete mathematics.

**Priority Definitions**

The following definitions are intended as a guideline to prioritize requirements.

* Priority 1 – anyone can access this application.
* Priority 2 – The requirement of the system is user is needed to connect to each other.
* Priority 3 – The requirement is ‘’lesson learn’’ give an idea to answer the question in this application.

It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category.

* A good requirement is:
* Correct
* Unambiguous (all statements have exactly one interpretation)
* Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
* Consistent
* Ranked for importance and/or stability
* Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)
* Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
* Does not specify any particular design
* Traceable (cross-reference with source documents and spawned documents).

## Functional Requirements

In the example below, the requirement numbering has a scheme - BR\_LR\_0## (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

For Example:

| Req# | Requirement | Comments | Priority | Days Completed | Date Rvwd | Reviewed / Approved |
| --- | --- | --- | --- | --- | --- | --- |
| BR\_LR\_05 | The system should associate a supervisor indicator with each job class. | Business Process = “Maintenance | 3 |  | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_08 | The system should handle any number of fees (existing and new) associated with unions. | Business Process = “Changing Dues in the System”  An example of a new fee is an initiation fee. | 2 |  | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_10 | The system should capture and maintain job class status (i.e., active or inactive) | Business Process = “Maintenance”  Some job classes are old and are no longer used. However, they still need to be maintained for legal, contract and historical purposes. | 2 |  | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_16 | The system should assign the Supervisor Code based on the value in the Job Class table and additional criteria as specified by the clients. | April 2005 – New requirement. It is one of three new requirements from BR\_LR\_03. | 2 |  |  |  |
| BR\_LR\_18 | The system should provide the Labor Relations office with the ability to override the system-derived Bargaining Unit code and the Union Code for to-be-determined employee types, including hourly appointments. | April 2005 – New requirement. It is one of three new requirements from BR\_LR\_04.  5/11/2005 – Priority changed from 2 to 3. | ~~2~~  3 |  |  |  |

## User Interface Requirements

User interface describe the function of output creating a button menu, option and sitting to required interface.

## Usability

Include any specific usability requirements, for example,

* Easy to learn
* The documentation will be the guide to the user.
* The help should be sensitive and explain how to play the game

## Performance

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

* The system numerical requirement is the number of users of a system; we make it for those interested in this system. It’s easy to use if the user is new.
* The number of system transaction will depending on the two options: single player and multiplayer, single player: each level has ten seconds to answer every question, if the question is not answered automatically the answer will be wrong. Then in multiplayer you can choose you want to play with you, there’s encrypted and decrypt this is the type of play in multiplayer**.**

### Capacity

The maximum capacity of the application is only two users in multiplayer then in the single player only one.

### Availability

Include specific and measurable requirements for:

* 1 hour per session of the class about discrete.
* They will think the answer
* quiz game , level difficulties and voice recognition.
* Impact of downtime on users and business operations

### Latency

Include our explanation the average needed is highest score .

## Manageability/Maintainability -SKIP

### Monitoring

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

### Maintenance

Specify attributes of the system that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

### Operations

Specify any normal and special operations required by the user, including:

* periods of interactive operations and periods of unattended operations
* data processing support functions
* backup and recovery operations
* safety considerations and requirements
* disaster recovery and business resumption

## System Interface/Integration -SKIP

Specify the use of other required products (e.g., a database or operating system), and interfaces with other systems (e.g., UWHires package interfaces with PubCookie and ODS, HEPPS system interfaces with Budget system). For each interface, define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

Outline each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

### Network and Hardware Interfaces

Specify the logical characteristics of each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

### Systems Interfaces

Example systems interface requirements:

1. System1-to-System2 Interface

The <external party> will create and send a fixed length text file as an email attachment to [System2mail@u.washington.edu](mailto:heppsmai@u.washington.edu) to be imported into the System2 system for payroll calculation. This file must be received on EDIT day by 4:00 PM in order to be processed in the EDIT night run. The requirements below document the file specifications, data transfer process, and specific schedule. This file is referred to as "FileName" in this document.

File Structure and Format

* 1. The FileName file is a fixed length text file.
  2. The FileName file is an unformatted ASCII file (text-only).
  3. The FileName file contains a batch totals record and several detail records.

File Description: Batch Totals Record

* 1. The batch totals record can be placed at the beginning, in the middle, or at the end of the file.
  2. The batch totals record contains the following:

Record Type (value: XA)

Process Type (value: A)

Batch Number (3 digit number assigned by Payroll Dept)

Origin Code (AIG)

Total number of detail records

Total deduction amount

File Description: Detail Records

* 1. The FileName file contains a row for each record meeting xxx criteria.
  2. Each row in the FileName file contains the following fields, comma-delimited and encased in double-quotes where the data includes commas or spaces:
* Employee Id
* Record Type
* Process Date (MMDDYY)
* XYG Number
* Element Code
* Amount
* Amount Sign
* Year Flag
* Total Amount
* Total Amt Sign

## Security

### Protection

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

* encryption
* activity logging, historical data sets
* restrictions on intermodule communications
* data integrity checks

### Authorization and Authentication

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie. Using Security Assertion Markup Language (SAML). How valid users can access the application, and avoid alteration of codes

## Data Management

Specify the requirements for any information that is to be placed into a database, including

* types of information used by various functions
* frequency of use
* data access rules
* data entities and relationships
* integrity constraints
* data retention
* valid range, accuracy, and/or tolerance
* units of measure
* data formats
* default or initial values

## Standards Compliance

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

## Portability

If portability is a requirement, specify attributes of the system that relate to the ease of porting the system to other host machines and/or operating systems. For example,

* Percentage of components with host-dependent code;
* Percentage of code that is host dependent;
* Use of a proven portable language;
* Use of a particular compiler or language subset;
* Use of a particular operating system;
* The need for environment-independence - the product must operate the same regardless of operating systems, networks, development or production environments.