

Functional Requirements Document

CALTOR

Version	Description of Change	Author	Date

CONTENTS

1	INTRODUCTION	4
1.1	Purpose.....	4
1.2	Scope.....	4
1.3	Background	4
1.4	References	4
1.5	Assumptions and Constraints.....	4
1.6	Document Overview	5
2	METHODOLOGY	5
3	FUNCTIONAL REQUIREMENTS	5
4.1	Context.....	5
4.2	User Requirements.....	6
4.3	Data Flow Diagrams	6
4.4	Logical Data Model/Data Dictionary.....	6
4.5	Functional Requirements	6
5	OTHER REQUIREMENTS	7
5.1	Interface Requirements	7
5.2	Data Conversion Requirements	7
5.3	Hardware/Software Requirements	8
5.4	Operational Requirements	8
	APPENDIX A - GLOSSARY	12

1 INTRODUCTION

The functional requirements document focuses on the requirements for upgrading a Grading Calculator for Academic transition program at the University of Regina. The document would include gathering information, calculation and analysis and would focus on function requirements, scope, system architecture and interface.

We software engineer students will deliver a system for students to have a proper and reliable calculator that is catered to their needs and requirements

1.1 Purpose

This document is for determining the behavior or the intended behavior of the grading calculator, description for the services that the software is offering, define the system and its components.

1.2 Scope

Grading Calculator System is basically updating the manual system that is currently used by SSW into an internet-based application so that users would have ease of access in accessing their accounts. It is specifically designed for SSW students since it will offer a customized interface catered specifically around them. The Grading Calculator System would be available for any students in SSW as well as the teachers / professors assigned to SSW.

1.3 Background

Normally, the SSW would rely on an excel sheet to manually input their grades then the existing excel template would generate the grades for them. Currently the excel template consists of many tabs and would let the users have a hard time to navigate and we plan to fix that with a user friendly interface to make it more convenient while also having data confidential and security in mind.

1.4 References

[List references and controlling documents, including: meeting summaries, white papers, other deliverables, etc.]

1.5 Assumptions and Constraints

This Document is based on the assumption that there would be no change in the process of calculating the grades, for the constraints, the system should be adapted to english, all the interfaces and also must be configurable.

1.5.1 Assumptions

ASP.NET Core has a number of features that we assume we could utilize for building our web application such as asynchronous patterns, their WEB API frameworks as well as their MVC since it is tailored for building web apps. The new feature called Razor Pages called also be utilized since it is an extension of MVC.

We can also assume that inputs would come from either touch screen keyboard or the computer keyboard and we would rely on this for applications navigation.

1.5.2 Constraints

One major constraint for the system would be the amount of users we can store in database also the web application might not be mobile friendly.

1.6 Document Overview

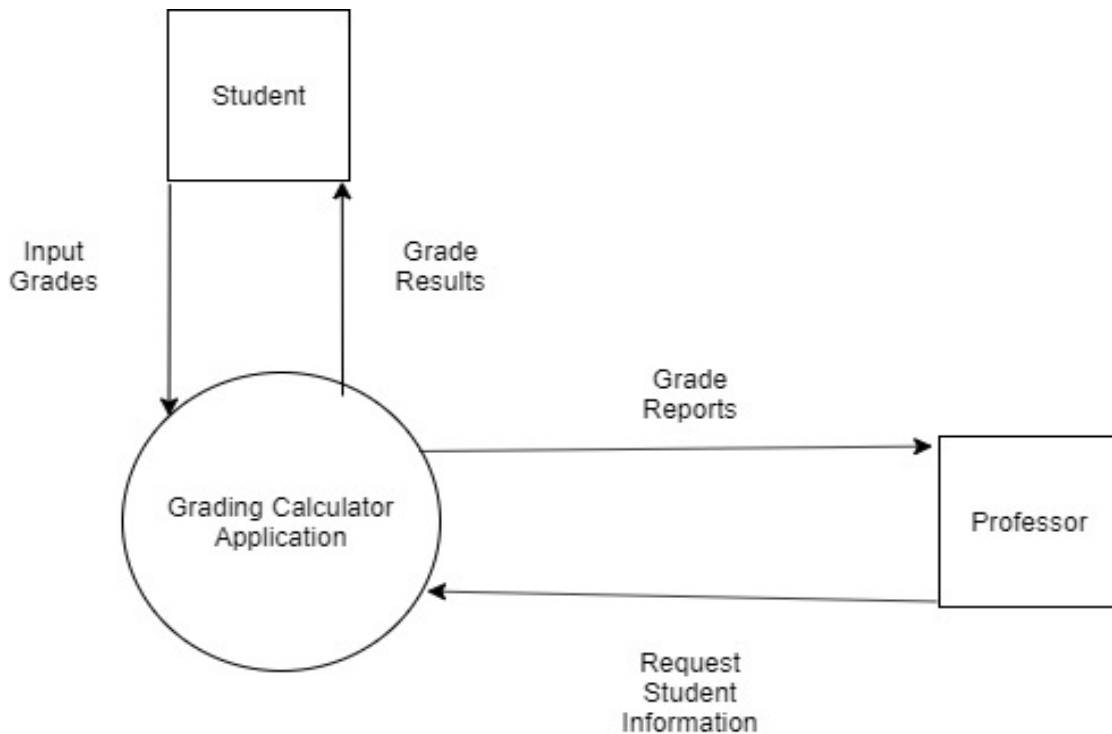
This section is a high level description of the system, which consists of constraints, assumptions, background, scope and purpose.

2 METHODOLOGY

[Describe the overall approach used in the determination of the FRD contents. Describe the modeling method(s) so non-technical readers can understand what they are conveying.]

3 FUNCTIONAL REQUIREMENTS

4.1 Context

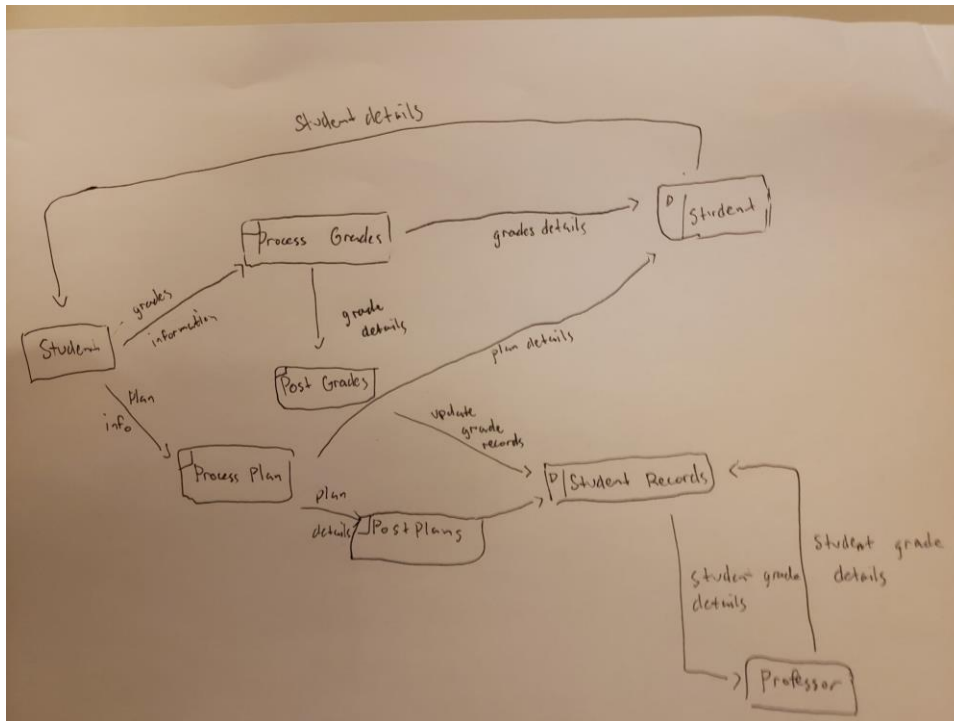


4.2 User Requirements

The Grading Calculator must be able to:

- Have user signup and log in
- Stores user's inputs (assignment grades, grades from previous classes)
- From those stored data be able to calculate the user's UGPA according the University of Regina standard
- Have those data and calculations saved when the user logs back in so that they don't have to keep entering their grades
- Have a different section for the Academic Transition Program class
- Be able to also be used by other students not just the ones in the program

4.3 Data Flow Diagrams



4.4 Logical Data Model/Data Dictionary

[Create the initial Logical Data Model. Describe data requirements by providing data entities, decomposition, and definitions in a data dictionary. The data requirements describe the business data needed by the application system. Data requirements do not describe the physical database and are not at the level of identifying field names.]

4.5 Functional Requirements

[List the functional requirements of the system.]

Students Group

Requirement ID	Requirement Definition
FR 1.0	The system shall prompt for user ID and password

FR 1.1	The system should allow for user to enter grades
FR 1.1.1	The system shall store data of user

Professors Group

FR 1.0	Requirement Definition
FR 1.1	The system shall prompt for user ID and password
FR 1.1.1	The system shall allow access to user database
FR 1.1.2	The system shall generate reports for students

5 OTHER REQUIREMENTS

[Describe the non-behavioral requirements.]

5.1 Interface Requirements

[Describe the user interfaces that are to be implemented by the system.]

5.1.1 Hardware Interfaces

[Define hardware interfaces supported by the system, including logical structure, physical addresses, and expected behavior.]

5.1.2 Software Interfaces

[Name the applications with which the subject application must interface. State the following for each such application: name of application, external owner of application, interface details (only if determined by the other application).]

It is acceptable to reference an interface control document for details of the interface interactions.]

5.1.3 Communications Interfaces

[Describe communications interfaces to other systems or devices, such as local area networks.]

5.2 Data Conversion Requirements

[Describe the requirements needed for conversion of legacy data into the system.]

5.3 Hardware/Software Requirements

[Provide a description of the hardware and software platforms needed to support the system.]

5.4 Operational Requirements

[Provide the operational requirements in this section.

Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be”? Do not state what steps will be taken to provide reliability.

Distinguish preferences from requirements. Requirements are based on business needs, preferences are not. If, for example, the user requires a special response but does not have a business-related reason for it, that requirement is a preference.

Other applicable requirements on system attributes may be added to the list of subsections below.]

Operational requirements describe how the system will run and communicate with operations personnel.

5.4.1 Security and Privacy

[Provide a list of the security requirements using the following criteria:

A. State the consequences of the following breaches of security in the subject application:

1. Loss or corruption of data
2. Disclosure of secrets or sensitive information
3. Disclosure of privileged/privacy information about individuals
4. Corruption of software or introduction of malware, such as viruses

B. State the type(s) of security required. Include the need for the following as appropriate:

1. Physical security.
2. Access by user role or types.
3. State access control requirements by data attribute. For example, one group of users has permission to view an attribute but not update it while another group of users has permissions to update or view it.
4. State access requirements based on system function. For example, if there is a need to grant access to certain system functions to one group of users, but not to another. For example, "The system shall make Function X available to the System Administrator only".
5. State if there is a need for certification and accreditation of the security measures adopted for this application]

The Security Section describes the need to control access to the data. This includes controlling who may view and alter application data.

5.4.2 Audit Trail

[List the activities recorded in the application's audit trail. For each activity, list the data recorded.]

5.4.3 Reliability

A. [State the following in this section:

1. State the damage can result from failure of this system—indicate the criticality of the software, such as:
 - a) Loss of human life
 - b) Complete or partial loss of the ability to perform a mission-critical function
 - c) Loss of revenue
 - d) Loss of employee productivity
2. What is the minimum acceptable level of reliability?

B. State required reliability:

1. Mean-Time-Between-Failure is the number of time units the system is operable before the first failure occurs.
2. Mean-Time-To-Failure is the number of time units before the system is operable divided by the number of failures during the time period.
3. Mean-Time-To-Repair is the number of time units required to perform system repair divided by the number of repairs during the time period.]

Reliability is the probability that the system processes work correctly and completely without being aborted.

5.4.4 Recoverability

[Answer the following questions in this section:

- A. In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?
- B. In the event the database is corrupted, to what level of currency must it be restored? For example “The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred”.
- C. If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?]

Recoverability is the ability to restore function and data in the event of a failure.

5.4.5 System Availability

[State the period during which the application must be available to users. For example, “*The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST.*” If the application must be available to users in more than one time zone, state the earliest start time and the latest stop time. Consider daylight savings time, too.]

Include use peak times. These are times when system unavailability is least acceptable.]

System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed.

5.4.6 General Performance

[Describe the requirements for the following:

- A. Response time for queries and updates
- B. Throughput
- C. Expected rate of user activity (for example, number of transactions per hour, day, or month, or cyclical periods)

Specific performance requirements, related to a specific functional requirement, should be listed with that functional requirement.

5.4.7 Capacity

[List the required capacities and expected volumes of data in business terms. Do not state capacities in terms of system memory requirements or disk space—if growth trends or projections are available, provide them]

5.4.8 Data Retention

[Describe the length of time various forms of data must be retained and the requirements for its destruction.

For example, “The system shall retain application information for 3 years”. Different forms of data include: system documentation, audit records, database records, access records.]

5.4.9 Error Handling

[Describe system error handling.]

5.4.10 Validation Rules

[Describe System Validation Rules.]

5.4.11 Conventions/Standards

[Describe system conventions and standards followed.]

For example: Microsoft standards are followed for windows, Institute of Electrical and Electronics Engineers (IEEE) for data formats, etc.]

APPENDIX A - GLOSSARY

[Define terms, acronyms, and abbreviations used in the FRD.]