

# **EduBuddy – Software Requirements Specification**

In Partial Fulfillment of the  
Requirements in CMSC 128:  
Software Engineering I

Third Laboratory for CMSC 128

Submitted by:  
Erru Torculas

Submitted to:  
Leonardo Pabroquez, Jr.  
**Faculty of Computer Science**

## Contents

<b>Contents</b>	<b>1</b>
<b>1. Introduction</b>	<b>2-4</b>
1.1 Purpose	2
1.2 Intended Audience	2
1.3 Intended Use	2
1.4 Scope	2-3
1.5 Definitions and Acronyms	3
1.6 Overview of the Document	4
<b>2. Overall Description</b>	<b>5</b>
2.1 User Needs	5
2.2 Assumptions and Dependencies	5
<b>3. System Features and Requirements</b>	<b>6-8</b>
3.1 Functional Requirements	6
3.2 External Interface Requirements	7
3.3 System Features	7
3.4 Nonfunctional Requirements	7-8
3.4.1 Performance	7
3.4.2 Safety	8
3.4.3 Security	8
3.4.4 Quality	8

## Introduction

### **1.1 Purpose**

The purpose of this document is to present a detailed version of EduBuddy. It will explain the purpose and features of the system, interfaces, functionalities, and constraints of the system. Not only that, this document's sole purpose is to guide the audience to fully navigate the details of the project and to assess the specifics needed for the features in order to develop.

### **1.2 Intended Audience**

This software requirements specification (SRS) document is intended for both the stakeholders and the developers of the system to fully visualize the requirements needed during the developing process. Apart from that, project managers will handle the whole team in leading the success of the system. Users will be included for an inclusive take on the project.

### **1.3 Intended Use**

This document entails specifics of the projects that will aid the developers in their process to create functionalities and constraints along the way. This will also be used by the stakeholders to oversee the business aspect of every requirements fitting to their standards. Lastly, project managers will use this document as a blueprint and the overview of what to expect of the team to develop.

### **1.4 Scope**

This software system will be an Educational Management Information System for students of Central Elementary School Municipality of Miagao. This system will be designed to maximize the student's productivity and learning capabilities by providing tools to assist them in automating course modules, educational videos, and messaging feature. By maximizing the student's learning efficiency and production, the system will meet the student's needs while remaining easy to grasp and use in the long run.

More specifically, the system is designed to allow instructors to manage and communicate with their students through the messaging feature of the system. The software will facilitate communication between instructors, students, and others. Also, this will enhance the student's efficiency by providing them.

### **1.5 Definitions and Acronyms**

<b>Term</b>	<b>Definitions</b>
Database	Collection of all the information monitored by the system.
Educational Management Information System	System for collecting, integration, processing, maintenance and dissemination of information to support decision-making , planning, monitoring and management in all levels of an education system
MERN	MERN stack comprises of MongoDB, Express, React, Node
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
SRS document	Software Requirements Specification document.
Stakeholder	Any person with an interest in the project who is not a developer.
User	Students or Instructor

### **1.6 Overview of Document**

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, System Features and Requirements section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

## **Overall Description**

### **2.1      *User Needs***

There will be a class of users that will have the permissions to access all functionality of the app, including:

- Creating channels where multiple users can interact.

- Deleting the channels
- Modifying and adding learning materials
- Allocating submissions for the students.
- Accessing educational materials.
- Authorization and verification of the users
- Users will have the freewill to edit their dashboards
- Calendars and tasks for submission updates

## **2.2      *Assumptions and Dependencies***

The software developed here assumes the use framework and technologies of stack MERN. These applications can be written in language for both server-side and client-side. All of the processes will be based on Windows operating system for portability purposes.

# **System Features and Requirements**

## **3.1      *Functional Requirements***

User story 1	As a student, I must be able to register and enter the website
<b><i>Functional Requirements</i></b>	
<b>1.1</b>	Login and Signup page

<b>1.2</b>	Logging in with Google or Facebook and its verification
<b>1.3</b>	Phone number verification
<b>1.4</b>	An “I forgot my password” section that sends a link to the verified email for retrieval of account.

<b>User story 2</b>	As an instructor, I must be able to edit or modify the course modules
<b><i>Functional Requirements</i></b>	
<b>2.1</b>	Editing, adding, and deleting different resources .or modules in the system

<b>User story 3</b>	As a student, I must be able to communicate with my instructor in case of help.
<b><i>Functional Requirements</i></b>	
<b>3.1</b>	Message through the web with options if video or call.
<b>3.2</b>	Automatic notifications if there’s a message
<b>3.3</b>	Phone number usage for communication purposes.

<b>User story 4</b>	As a student, I must be able to view my progress and tasks due on a specific date.
<b><i>Functional Requirements</i></b>	
<b>4.1</b>	Student’s dashboard page
<b>4.2</b>	A page with progress bars and calendars for visualization purposes only.

### **3.2 External Interface Requirements**

The only link to the external system is the link to EduBuddy database to verify the identification of users. MongoDB or MySQL will meddle the whole verification phase of the system in which the fields of interest to EduBuddy are user’s name, school/faculty (ID) number, email address, and password. Apart from that, MongoDB or MySQL will be used for the database where the fields of interest to EduBuddy are the user’s name, school/faculty (ID) number, email address, and password.

### **3.3      *System Features***

The software will have various features to fully operate its use. As for the authorization of the users, will verify their identification using the Login/Signup system.

The course module will be the main component of the system that the users will have to navigate.

### **3.4      *Nonfunctional Requirements***

#### **3.4.1   *Performance***

The EduBuddy will be on a server with high-speed internet capability. The physical machine to be used will be determined by the school during the prototyping phase. The application should load and be usable within 3 seconds. Also, the update of the interface on interaction should only be 2 seconds. The database should be normalized to prevent redundant data to improve performance. Importantly, the database should be distributed to prevent outages.

#### **3.4.2   *Safety***

Databases such as MongoDB or MySQL will meddle the whole security of the users were storing data is crucial for the development of the project. Backups of these databases should be done hourly and be kept for one week.

#### **3.4.3   *Security***

The server on which the educational modules reside will have its own security to prevent unauthorized overwriting access. Apart from that, this will mitigate the copyright issues and intellectual infringement of the instructors and the school per se. There is no restriction on reading access. The use of school email by the users is on the client systems and thus is external to the system. Databases should be behind a firewall.



The PC on which the admin resides will have its own security. Only they will have physical access to the machine and a separate program for monitoring purposes. There is no special protection built into this other than to provide the admin to oversee the access of educational modules.

#### 3.4.4 *Quality*

- Availability: This application is critical to the educational system, we will have a goal of four nines(99.99%) availability.
- Correctness: The application should never allow duplication of the user. Unauthorized access of the account not intended by the user is prohibited
- Maintainability: The application should use continuous integration so that features and bug fixes can be deployed quickly without downtime
- Usability: The interface should be easy to learn and intuitive for the users to navigate without a tutorial. Also, allowing users to accomplish their goals without errors.