SOFTWARE REQUIREMENT SPECIFICATION – SRS.

Website AI Chatbot supports English learning for online

learning websites.

Version 1.0

TABLE OF CONTENT

Contents

[Introduction 3](#_Toc178748958)

[1.1 Scope 3](#_Toc178748959)

[1.2 Terminology 3](#_Toc178748960)

[1.3 References 3](#_Toc178748961)

[1.4 Overview 3](#_Toc178748962)

[User requirements 4](#_Toc178748963)

[2.1 Stakeholder 4](#_Toc178748964)

[2.2 System features 4](#_Toc178748965)

[2.3 Overview use case diagram 4](#_Toc178748966)

[2.3.1Use case decomposition diagram 4](#_Toc178748967)

[2.3.2 Decompose the “Administrator” use case 4](#_Toc178748968)

[2.3.3. Decompose the use case “Lecturer” 4](#_Toc178748969)

[2.5 Business Process 4](#_Toc178748970)

[2.5.1 Software Usage Process 4](#_Toc178748971)

[2.5.2 Instructor Management Process 4](#_Toc178748972)

[2.5.3 Course Management Process 4](#_Toc178748973)

[2.5.4 Instructor Management Process 4](#_Toc178748974)

[2.5.5 Student Management Process 4](#_Toc178748975)

[2.5.6 Notifications management process 4](#_Toc178748976)

[2.5.7 FAQ Management Process 4](#_Toc178748977)

[2.5.8 Student's software usage process 4](#_Toc178748978)

[2.6 Specification of use-cases 4](#_Toc178748979)

[2.6.1 Login 4](#_Toc178748980)

[2.6.2 Change password 4](#_Toc178748981)

[2.6.3 Register 4](#_Toc178748982)

[2.6.4 Personal information update 4](#_Toc178748983)

[2.6.5 Search for news, FAQs, courses, course history, lectures 4](#_Toc178748984)

[2.6.6 Intructor management 4](#_Toc178748985)

[2.6.7 Course management 4](#_Toc178748986)

[2.6.8 Student management 4](#_Toc178748987)

[2.6.9 Lecture Management 4](#_Toc178748988)

[2.6.10 Notification management 4](#_Toc178748989)

[2.6.11 FAQs management 4](#_Toc178748990)

[2.6.12 View course history and student information 4](#_Toc178748991)

[2.6.13 Course Category Management 4](#_Toc178748992)

[2.6.14 Using the system function of the student 4](#_Toc178748993)

[Non-functional requirements 5](#_Toc178748994)

[3.1 User interface 5](#_Toc178748995)

[3.2 Security 5](#_Toc178748996)

[3.3 Security 5](#_Toc178748997)

FIGURE

# Introduction

## 1.1 Scope

## 1.2 Terminology

## 1.3 References

## 1.4 Overview

# User requirements

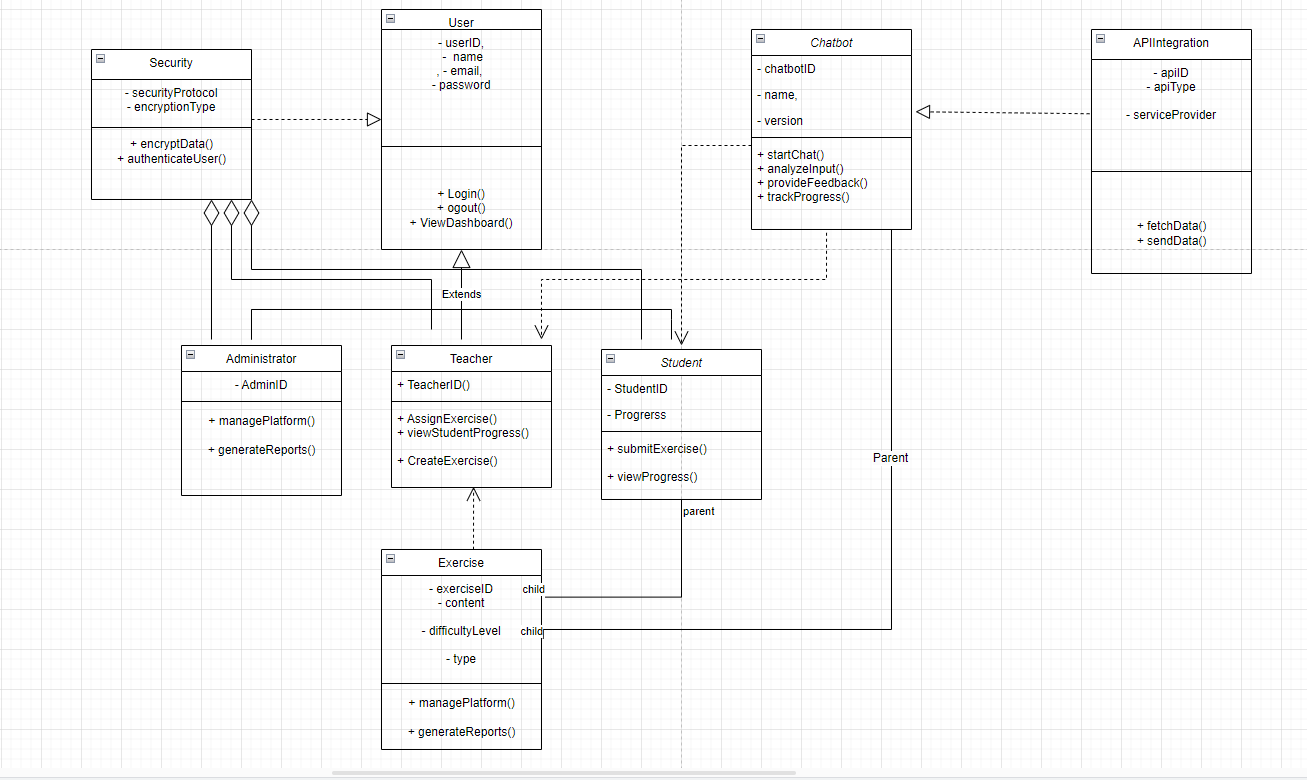
## 2.1 Stakeholder

## 2.2 System features

## 2.3 Overview use case diagram

## 2.3.1Use case decomposition diagram

**Class Diagram**



**Explain class Diagram**

**1. User Class:**

* **Attributes**:
  + userID
  + name
  + email
  + password
* **Methods**:
  + login()
  + logout()
  + viewDashboard()
* **Relationships**:
  + Inheritance relationship with Administrator, Teacher, and Student classes (they extend from User).

**2. Administrator Class (extends User):**

* **Attributes**:
  + adminID
* **Methods**:
  + managePlatform()
  + generateReports()
* **Relationship**:
  + Inherits from User.

**3. Teacher Class (extends User):**

* **Attributes**:
  + teacherID
* **Methods**:
  + assignExercise()
  + viewStudentProgress()
  + createExercise()
* **Relationship**:
  + Inherits from User.

**4. Student Class (extends User):**

* **Attributes**:
  + studentID
  + progress
* **Methods**:
  + submitExercise()
  + viewProgress()
* **Relationship**:
  + Inherits from User.

**5. Exercise Class:**

* **Attributes**:
  + exerciseID
  + content
  + difficultyLevel
  + type
* **Methods**:
  + managePlatform()
  + generateReports()
* **Relationships**:
  + Has a parent-child relationship with both Teacher (Teacher creates exercises) and Student (Student submits exercises).

**6. Security Class:**

* **Attributes**:
  + securityProtocol
  + encryptionType
* **Methods**:
  + encryptData()
  + authenticateUser()
* **Relationship**:
  + Association with the User class, likely providing security services for login and authentication.

**7. Chatbot Class:**

* **Attributes**:
  + chatbotID
  + name
  + version
* **Methods**:
  + startChat()
  + analyzeInput()
  + provideFeedback()
  + trackProgress()
* **Relationship**:
  + Communicates with the User and possibly Student for tracking progress or assisting in the platform.

**8. API Integration Class:**

* **Attributes**:
  + apiID
  + apiType
  + serviceProvider
* **Methods**:
  + fetchData()
  + sendData()
* **Relationship**:
  + Associated with Chatbot, suggesting the chatbot uses APIs to fetch or send data.

**Key Relationships:**

* **Inheritance**:
  + Administrator, Teacher, and Student inherit from the User class.
* **Parent/Child**:
  + Exercise is related to Teacher and Student.
* **Association**:
  + Security class is linked to User (handles authentication).
  + APIIntegration is linked to Chatbot

## 2.3.2 Decompose the “Administrator” use case

## 2.3.3. Decompose the use case “Lecturer”

## 2.5 Business Process

## 2.5.1 Software Usage Process

## 2.5.2 Instructor Management Process

## 2.5.3 Course Management Process

## 2.5.4 Instructor Management Process

## 2.5.5 Student Management Process

## 2.5.6 Notifications management process

## 2.5.7 FAQ Management Process

## 2.5.8 Student's software usage process

## 2.6 Specification of use-cases

## 2.6.1 Login

## 2.6.2 Change password

## 2.6.3 Register

## 2.6.4 Personal information update

## 2.6.5 Search for news, FAQs, courses, course history, lectures

## 2.6.6 Intructor management

## 2.6.7 Course management

## 2.6.8 Student management

## 2.6.9 Lecture Management

## 2.6.10 Notification management

## 2.6.11 FAQs management

## 2.6.12 View course history and student information

## 2.6.13 Course Category Management

## 2.6.14 Using the system function of the student

# Non-functional requirements

## 3.1 User interface

## 3.2 Security

## 3.3 Security