Flexible Virtual School

Phase 1: Problem Definition

Contributors

- **Talal Alothman**
- **Mshari Alaeena**
- **Saud Alkatheeri**
- **Saleh Alghaith**
- **Khaled Alharbi**
- **Talal Alkahtani**

Table of Contents

Introduction 3

Project Description 3

Purpose 3

Problem Domain Analysis 3

Problems to be addressed 3

Addressing the Problems: 4

Concepts and Vocabulary 4

System Context View 5

1st scenario: Student Application for Diploma. 5

2nd scenario: Instructor conducts an exam. 7

Functions & NFP of the System 9

Functional properties 9

Student 9

Instructor 9

Admin 10

System 10

Non-Functional properties 10

Reliability 10

Security 10

Portability 10

Performance 10

Usability 10

Scalability 10

Challenges 11

Projection 11

Revision Control History

Version	Date	Description of change

Introduction

Project Description

The project involves the design and build of a virtual school that offers programs which aim to teach the Arabic language to the native Arabic and non-Arabic native people via modern internet technologies such as virtual life, second life, and web services.

Purpose

The purpose of this project is to make the programs of the academy more accessible and effective. By building a virtual school, the academy aims to attract more students, offer flexible courses, and optimize resources.

Our involvement in this project will lead to professional growth and skill development in the field of online education and what drives us to work on this project is to make the learning of the Arabic language reachable from all over the world.

Problem Domain Analysis

Domain description: The problem domain focuses on the educational sector, specifically teaching the Arabic language. The Arabic Universal Academy would like to build a virtual school. The domain includes language education for both native Arabic speakers and non-Arabic speakers. The goal is to use modern internet technologies such as virtual life, second life, and web services, to create a flexible and accessible platform for teaching Arabic worldwide.

Problems to be addressed

1. **Limited Accessibility:** Traditional approaches limit the academy's reach, constraining its ability to attract more students.
2. **Inflexibility:** Traditional education methods often lack flexibility in terms of course scheduling, locations, and instructor choices.
3. **Integrating Technology:** Integrating new internet technologies, like virtual life and web services, into the educational platform.
4. **Resource Optimization:** The need to optimize resources, including instructors and facilities, to provide for a larger and more diverse student population.

Addressing the Problems:

1. Designing and building a virtual school to provide a modern and accessible platform for language education.
2. Offering courses that are flexible in terms of timing, location, and instructor choices.
3. Collaborating with experts to integrate new internet technologies into the virtual school infrastructure.
4. Provide training sessions for existing faculty members to enhance their skills in online teaching methodologies.

Concepts and Vocabulary

Virtual School: An educational institution that offers diplomas and programs primarily through online platforms.

Diploma: Structured set of programs designed to achieve the required learning objectives and outcomes, leading to the attainment of a diploma certificate upon successful completion.

Program: structured set of courses and activities designed to achieve specific educational objectives and outcomes within a particular subject area such as speaking, writing, grammar, and listening.

Web Services: Online technologies and platforms that help with the delivery of educational content, communication, and administrative functions over the internet.

Second Life: A 3D virtual world where students and instructors interact in real-time, attending and conducting classes online, as if they were in a physical classroom, but through digital avatars.

System Context View

1st scenario: Student Application for Diploma.

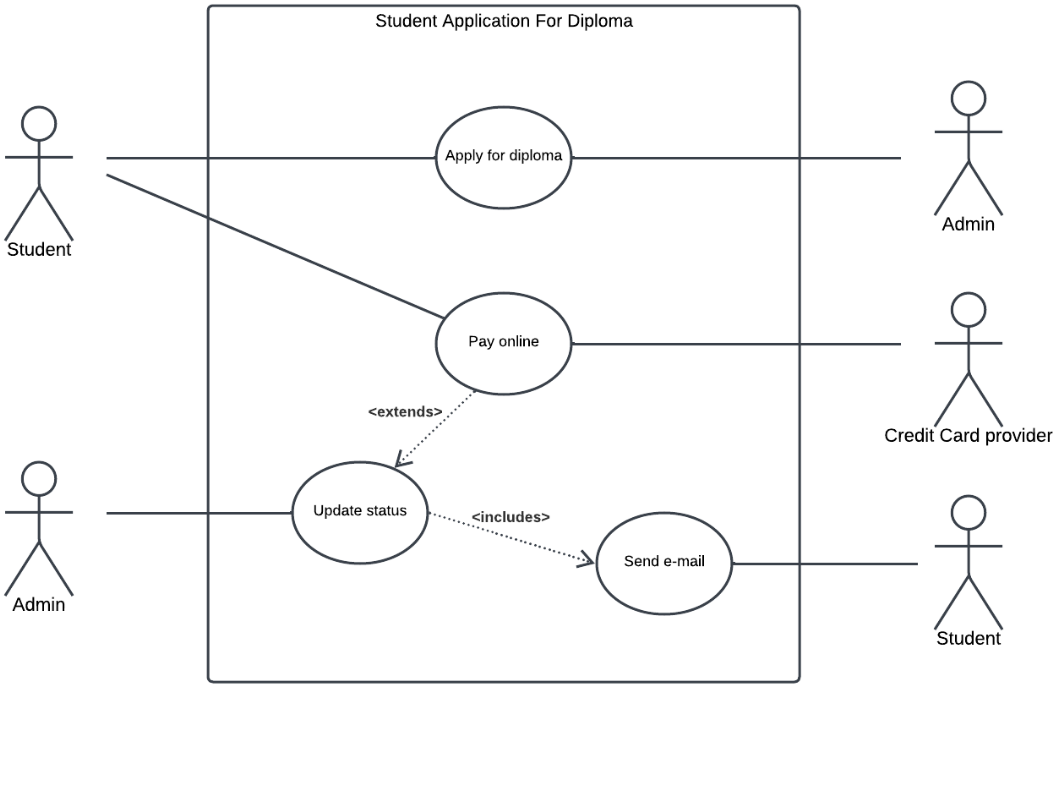
Description: A student chooses a specific diploma he wishes to apply for, after that fills out the application document then submits it throughout the system, the admin views the application and decides whether to accept or reject. In case of acceptance, the system sends an invoice through e-mail to the student to pay the required fees. If payment was successful, the system enrolls that student into the diploma program.

Expected inputs: A filled diploma application document by the student to be processed by the system.

System outcomes: A "Check application status" feature and an automated notification will be sent to keep both the student and admin informed about the status of application, acceptance decision, payment requests, and enrollment confirmation.

Benefits: Enhanced User Experience: By providing a user-friendly interface for the student to apply, pay fees, and track their enrollment status, the system enhances the overall experience for both students and admins.

Use case diagram.



Use Case Description	
System: *Flexible virtual school*	
Use Case name: *Apply for diploma*	
Primary actor: *Student*	**Other actors:** *Admin*
Stakeholders:	
Description: *The student will apply for a diploma program.*	
Relationships **▪Includes:** **▪Extends:**	
Input: filled diploma application document.	
Pre-conditions: *Student logged in.*	
Steps:	
Actor	**System**
1. Chooses a diploma program. *3. Fills the application form.*	*2. Displays the application form.* *4. Verifies the information.* *5. Notifies the admin.*
Alternative and exceptional flows: *4.1 IF there is missing information, THEN ask the student to fill in the missed part.*	
Post-conditions:	

2nd scenario: Instructor conducts an exam.

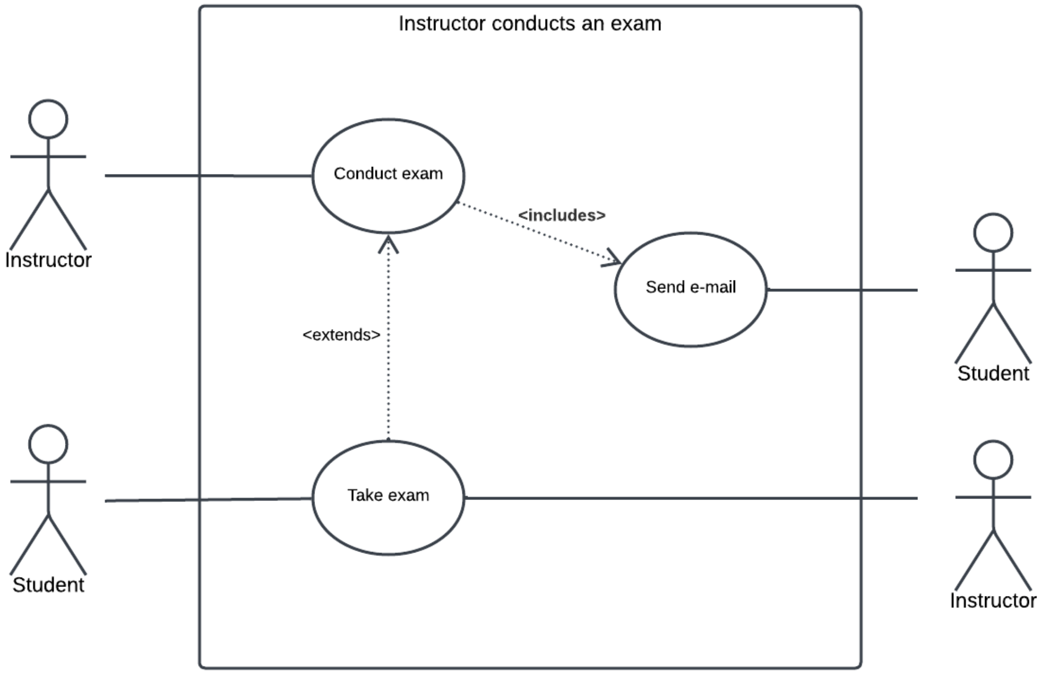
Description: The instructor begins by setting up the exam within the system through an exam creation tool, specifying details. Once the exam is set up, the instructor publishes it within the course, students enrolled in the course will be informed. On the designated date and time, students log into the system and access the exam through the course page.

Expected inputs: instructor specifies exam duration, format, and questions.

System outcomes: Automated notifications or announcements will be sent to the students informing them about the availability of the exam, along with any relevant instructions.

Benefits: The system automates exam administration, including exam setup, distribution, and grading, saving instructors time and effort.

Use case diagram



Use Case Description	
System: *Flexible virtual school*	
Use Case name: *Conduct exam*	
Primary actor: *Instructor*	**Other actors:**
Stakeholders:	
Description: *The instructor will conduct an exam.*	
Relationships ▪**Includes:** *send e-mail* ▪**Extends:** *take exam*	
Input: exam duration, format, and questions.	
Pre-conditions: *Instructor logged in.*	
Steps:	
Actor	**System**
1. Chooses a specific course. *3.Command: Create exam.* *5. Fills the exam details.* *6. Command: Save updates.*	*2. Displays the course page.* *4. Displays the exam creation tool.* *7. Saves updates.* *8. Sends e-mail to students.*
Alternative and exceptional flows:	
Post-conditions: *Student notified.* *Exam published.*	

Functions & NFP of the System

Functional properties

Student

1. The student shall be able to register.
2. The student shall be able to log-in using username and password.
3. The student shall be able to apply for a diploma.
4. The student shall be able to enroll in a program.
5. The student shall be able to view the contents of programs and diplomas.
6. The student shall be able to view course details (time and Instructor).
7. The student shall be able to search for courses based on time and instructor.
8. The student shall be able to attend virtual classes from any location.
9. The student shall be able to submit assignments.
10. The student shall be able to participate in discussions.
11. The student shall be able to take exams.
12. The student shall be able to view their progress.
13. The student shall be able to access course materials[1].
14. The student shall be able to give feedback about the instructor.
15. The student shall be able to pay online by credit card.
16. The student shall be able to request a refund for any corresponding fees paid.
17. The student shall be able to view their enrolled programs.
18. The diploma student shall be to view their enrolled semesters.
19. The diploma student shall be able to delete a semester from their enrollment.

Instructor

1. The instructor shall be able to register.
2. The instructor shall be able to log-in using username and password.
3. The instructor shall be able to conduct virtual classes.
4. The instructor shall be able to upload course materials.
5. The instructor shall be able to assess student performance.
6. The instructor shall be able to assign exams.
7. The instructor shall be able to manage [2] exams.

[1] materials: recorded lectures, syllabus, tutorials, and assignments.

[2] manage: add, delete, view, and edit.

Admin

1. The admin shall be able to manage [3] users' accounts.
2. The admin shall be able to manage course offerings.
3. The admin shall be able to update status[4] of diploma applications.
4. The admin shall be able to manage programs.

System

1. On update status, the system shall send an email to the applicant.
2. On applying for diploma, the system shall notify the admin.
3. On refund request, the system shall notify the student of the status of their refund request.

Non-Functional properties

Reliability

1. Major failures shall not exceed 0.4 per month.
2. The system shall be available 98% of the year.

Security

1. Online payments shall conform to the security standard of PCI DSS.
2. All the internet communications in the system shall be encrypted.
3. The instructor shall be able to conduct a virtual class at most 10 minutes before the class time.

Portability

1. The system shall be compatible with MacOS, Windows, Linux, Android, and IOS.

Performance

1. All the materials in the system shall be downloadable in less than 5 seconds.
2. The system shall achieve an average response time of no more than 2 seconds for user interactions, such as loading pages, accessing course materials, and submitting assignments and exams.

Usability

1. User shall be able to learn to use the system in 2 hours.
2. Average time of applying for a diploma shall be no longer than 15 minutes.

Scalability

1. The system shall be able to accommodate 100 users simultaneously in the video conferencing mode.
2. The database infrastructure shall be scalable to support the storage and processing needs of a growing user base.

[3] manage: add, delete, view, and edit.

[4] update status: accept, reject and, under review.

Challenges

Integration with Existing Systems

Integrating the virtual school platform with existing systems, such as payment gateways, and content management systems.

Security and Privacy

Keeping sensitive student data, ensuring secure access to online resources, and protecting against cyber threats such as hacking, phishing, and data breaches are critical considerations for our system.

Customization and Flexibility

Allowing flexibility in scheduling, content delivery, and instructor-student interactions.

Managing Scalability and Performance

Ensuring high performance and responsiveness of the virtual school platform, especially during peak usage times.

Training and Support

Providing adequate training and support resources for students and staff helps them navigate the system effectively.

Projection

By the end of the semester, our objectives include acquiring the ability to design user-friendly interfaces, utilizing software design tools proficiently for creating comprehensive software system architectures, and gaining knowledge about the science of design in the realm of software engineering. The time limit will be optimally utilized to enhance performance and develop skills in these key areas.

List of skills or knowledge expected: -

- Analyzing an engineering problem and selecting the most suitable design solution is a crucial skill.
- Designing software systems with the relevant architectural styles is a key competency.
- Understanding the connection between a system's non-functional requirements and its design goals is essential.