SOFE 3650-Fall 2022

Software Design and Architectures

Project ADD Iteration 2 & 3

Date: December 6th, 2022

Rayan Pasha - 100704599
Andy Lau - 100754380
Mohammed Khan - 100756901

Use Cases

Use Case	Description
UC1: Course registration	Courses are displayed to the user, and allowed to filter for or search for courses. Upon successful registration, the system database is updated.
UC2: Check grades for registered courses	Students are able to make a request to the system database to check their grades for their current courses.
UC3: Subscribe to exams	Students are able to subscribe to their exams
UC4: Private login	A user logs into the system through a login/password screen. Upon successful login, the user is presented with different options
UC5: Edit Grades	The lecturer adds, removes or edits grades from the system database.
UC6: Schedule an exam	Lecturer adds or edits the date for an exam in the system database.
UC7: Downtime warning	Maintainer inputs a message through the system that is displayed to all users to inform them of the system going down temporarily.
UC8: Add new course(s)	Administration adds, edits or removes a course or courses from the system database.

Quality Attributes

ID	Quality Attribute	Scenario	Associated Use Case
QA-1	Availability	If the system fails during normal operation then log the fault and resume operations in 1 second.	All
QA-2	Usability	If the user wants to change the system language to Dutch or English during normal operations. They will be able to do so by clicking a button and the system will change language in 3 seconds.	All

QA-3	Security, Performance	If a student logins into the system, it returns a student profile UI with no downtime.	UC4: Private Login
QA-4	Security, Performance	A user performs a change in system data during normal operations. It is possible to know who performed the operation and when it was performed 100% of the time. System data is changed within 24 hours	UC1:Course Registration UC3: Subscribe to Exams UC5: Edit Grades UC6: Schedule an Exam
QA-5	Interoperability, Performance	A student requests data from the system during normal operations. The system sends the data to the student and continues to operate with no downtime.	UC2: Check Grades For Registered Courses
QA-6	Modifiability	It is expected that new students, lecturers, and courses will be added to the system in the future. They should be able to be added to the system with minimal changes.	UC2: Check Grades For Registered Courses UC1:Course Registration

Constraints

ID	Constraint
Con-1	Students that do not have prerequisite courses should not be able to register for the related course
Con-2	The system must be able to run on different types of browsers and devices
Con-3	Only the administration, system database and, the system maintainer are able to change the data inputs of the course management system
Con-4	Must support at least 100 users using the system at the same time
Con-5	The system must be up and running for the majority of the year. (2 week yearly maintenance)
Con-6	A unique and specific database server must be created and used for the course management

	system and for the course management system alone.

Concerns

ID	Concerns
CRN-1	Maintain the privacy and security of the data within the system
CRN-2	A experienced team must be made for the development and maintenance of the course management system
CRN-3	Creating a functional and comprehensive initial structure for the course management system to be developed on

Step 2: Establish Iteration Goal by Selecting Drivers

The primary use cases for the course registration system are:

- UC1: Course registration
- UC4: Private login
- UC5: Edit Grades

Step 3: Choose One or More Elements of the System to Refine

There are multiple elements of the system that we would like to refine, starting with expanding upon the security system. We would like to make all users feel comfortable and safe when using our course management system. We would like to eliminate the security risks associated with the system. Only users should have access to their own grades and courses with the exception of academic administration. We would like to implement a 2 step authentication process in the later stages of creating the system providing a more secure connection for not only the students but for professors and other forms of administration.

Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers

Design Decisions and Location	Rationale and Assumptions
Grade data should be modifiable	Allows Lecturers to change grades data in order to fulfill (UC5: Edit Grades).
Decompose elements from iteration 1 in domain specific components	Helps in identification of major entities that reside in the layers

Use HTML, CSS, and JavaScript	HTML, CSS, and javascript are industry standard languages that the development team is already familiar with which will lead to greater productivity.
Decompose domain objects into specialized components	Create components for objects in order to give functionality to the object

Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

Design Decisions and Location	**Rationale**
Create dedicated module for logging in	Login functionality is a big requirement for the project therefore we shall dedicate an entire module for it. (UC4: Private login)
Create dedicated object for grades	Grades are an important aspect for our system so we should devote an entire module for course functionality (UC1: Course registration) (UC8: Add new course(s))
Create dedicated module for editing grades	Ensures Lecturers can edit student grades (UC1: Course registration)
Decompose elements into rich client reference model into domain specific modules	This ensures all modules will support all functionalities that have been identified. This technique will be done for all use cases. We will use the noun/verb identification technique to decide if these are domain components or entities.

Step 6: Sketch Views and Record Design Decisions

Analysis of Use Cases for key entities

Use Case	**Description**

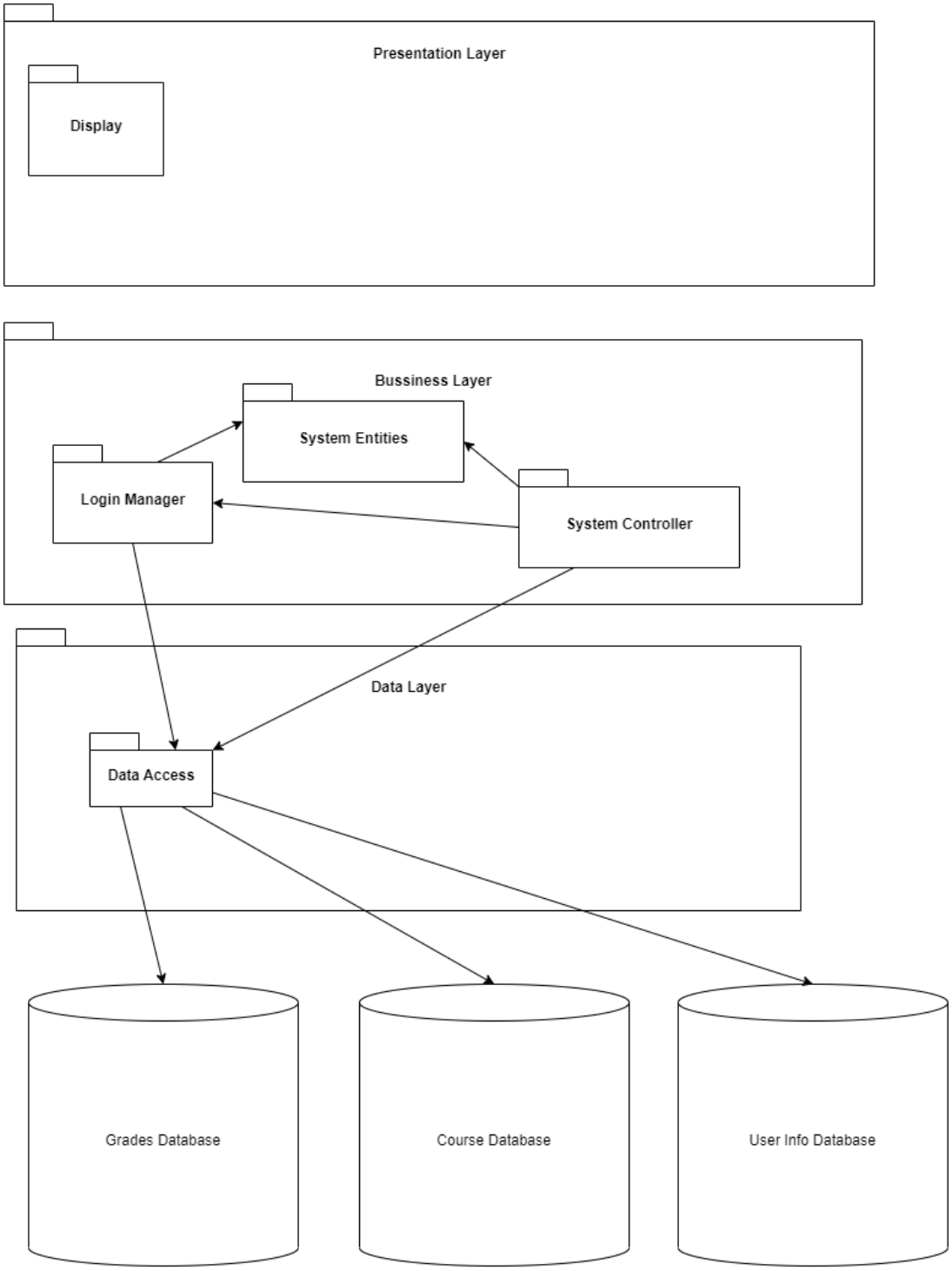
UC1: Course registration	Courses are displayed to the user, and allowed to filter for or search for courses. Upon successful registration, the system database is updated.
UC2: Check grades for registered courses	Students are able to make a request to the system database to check their grades for their current courses.
UC3: Subscribe to exams	Students are able to subscribe to their exams
UC4: Private login	A user logs into the system through a login/password screen. Upon successful login, the user is presented with different options
UC5: Edit Grades	The lecturer adds, removes or edits grades from the system database.
UC6: Schedule an exam	Lecturer adds or edits the date for an exam in the system database.
UC7: Downtime warning	Maintainer inputs a message through the system that is displayed to all users to inform them of the system going down temporarily.
UC8: Add new course(s)	Administration adds, edits or removes a course or courses from the system database.

Modified version of Rich Client architecture based on step 5.

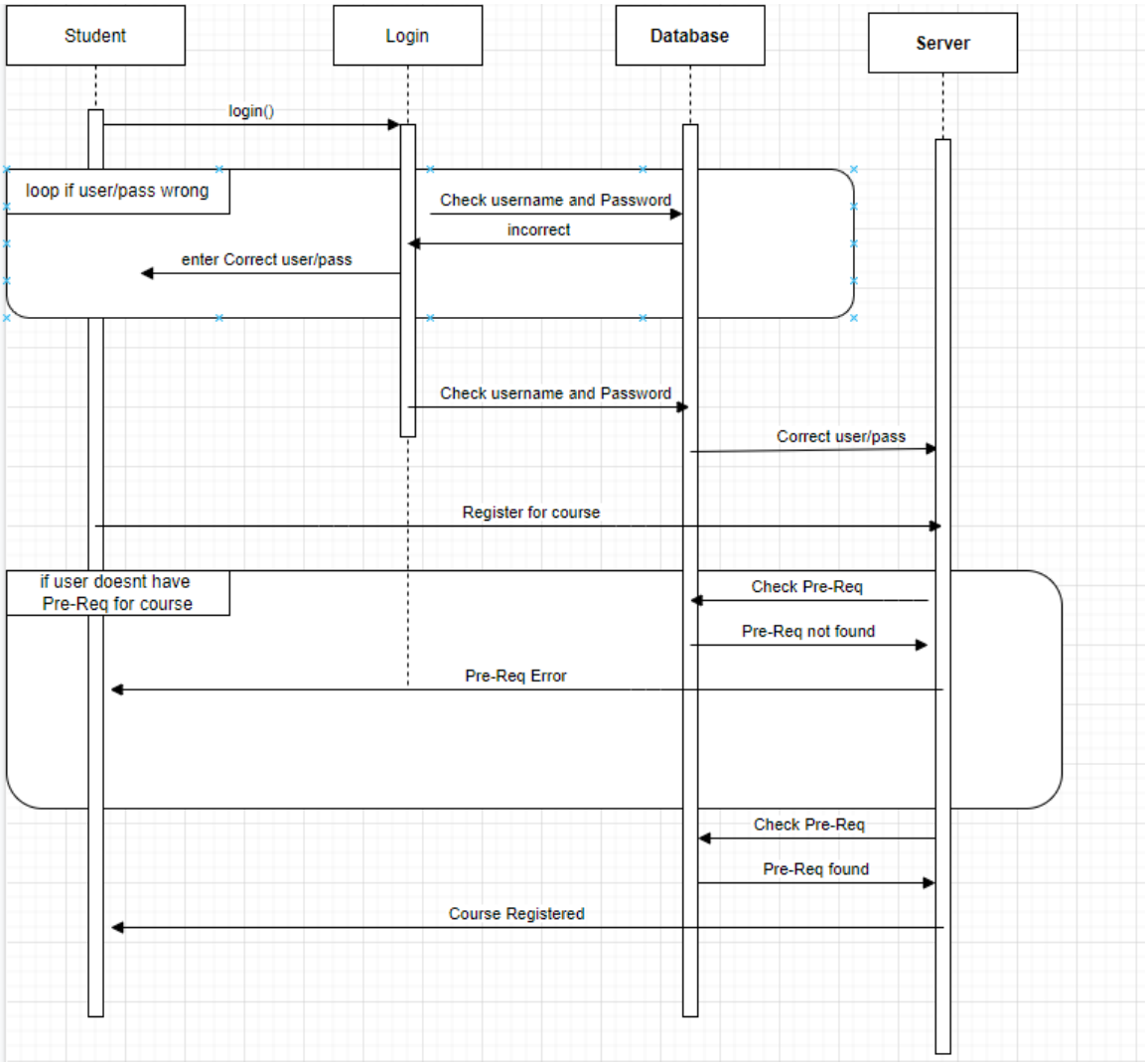
Element	**Domain Component**	**Domain Entity**
Courses	This entity will be a property of the Database.	
System		Stand-alone entity
Database		Stand-alone entity
Exams	This entity will be a property of the Database.	
Grades	This entity will be a property of the Database.	
Message	Part of interface to Presentation Layer	

Screen	Interface to the display at the Presentation Layer	

Logical architecture diagram



Element	Responsibility
Login	The login gives the students access to their grades, exam booking and exam schedules. The login must be secure so that only students with their own corresponding login information can log into their accounts.
Database	The database stores all the data for the system (grades, exam times, exam bookings etc) so that the website can call the information when it needs to be viewed.
Server	The server is responsible for making the website accessible to everyone by allowing the users to access it through the internet



Sequence diagram for Use Case 1 and 4 along with constraint 1. The sequence diagram states the action of a student logging in and adding a course to his classes next semester. The diagram loops when the student enters the wrong username or password to make sure they login to their own account. The second loop checks the database for the pre requisite courses that the student might need to register for the new class.

Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design

Not Addressed	**Partially Addressed**	**Completely Addressed**	**Design Decisions Made During the Iteration**
		UC1	Module essential for the course management system to work correctly
		UC4	Module needed for all users crucial for privacy and security
		UC8	Modules across the layers and preliminary interfaces to support this use cases have been identified.
	QA1		The elements that support the associated use case (UC-7) have been identified.
	QA2		No relevant decisions made.
	QA3		The elements that support the associated use case (UC-4) have been identified.

	QA4		The elements that support the associated use case (UC-1, UC-3, UC-5, UC-6) have been identified.
	QA5		The elements that support the associated use case (UC-2) have been identified
	QA6		The elements that support the associated use case (UC-2, UC-1) have been identified
	CRN-1		Module associated with UC4 and deals with the security aspect of the system
	CRN-2		No relevant decisions made.
	CRN-3		No relevant decisions made.

Project Iteration 3

Step 2: Establish Iteration Goal by Selecting Drivers

The primary use cases for the course registration system are:
- UC2: Check grades for registered courses
- UC3: Subscribe to exams
- UC6: Schedule an exam

Step 3: Choose One or More Elements of the System to Refine

We would like to refine the grades and exams system in order to make the lives of the students and lecturers easier. The grades and exams page should be easy to use and access. Lecturers should be the only ones allowed to book an exam and students should be the only ones allowed to subscribe to exams.

Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers

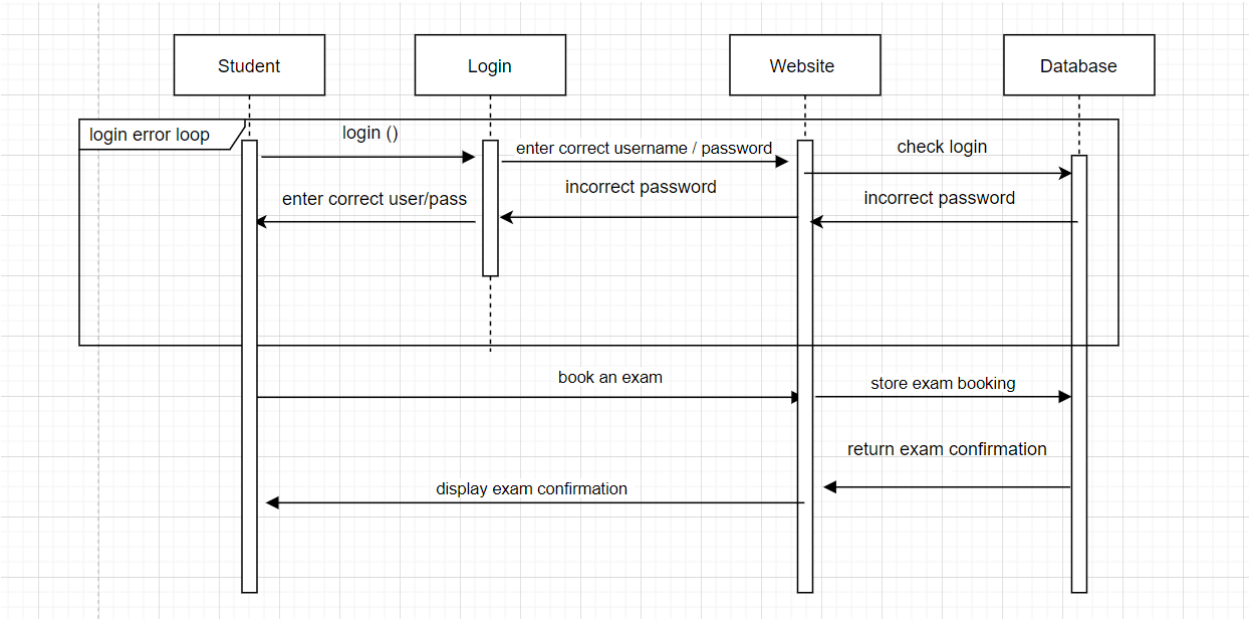
Design Decisions and Location	Rationale and Assumptions
Exam page access should allows be accessible after logging in	Allows access to exams (UC6: Schedule an exam) (UC3: Subscribe to exams) and ensures security (CRN-1)

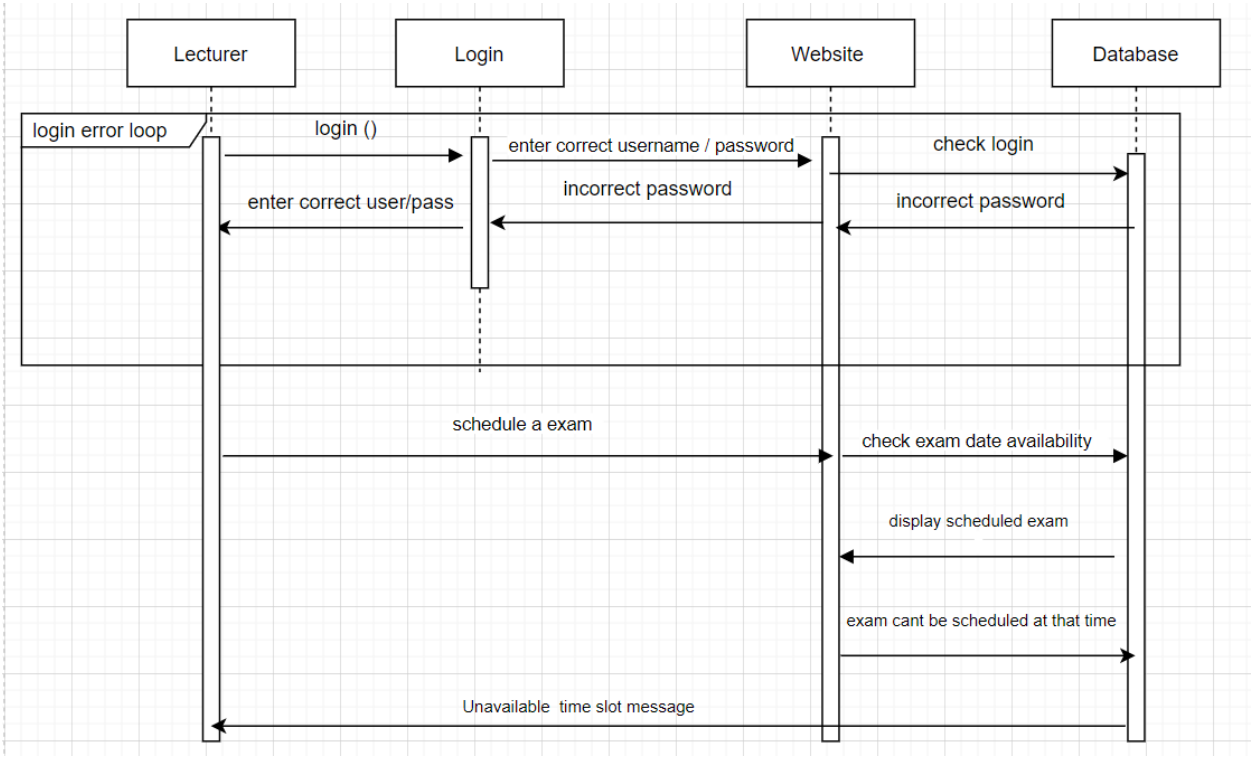
User login info is stored in a database	User data is secured and protected (CRN-1) (UC4: Private login)

Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

Design Decisions and Location	**Rationale**
Dedicated Exam Module	Facilitates the subscription to and scheduling of exams (UC6: Schedule an exam) (UC3: Subscribe to exams)
Dedicated grade page view module	Facilitates the viewing of grades (UC2: Check grades for registered courses)

Step 6: Sketch Views and Record Design Decisions





Element	Responsibility
Login	The login gives the students and lecturers access to their grades, exam booking and exam schedules. The login must be secure so that only students with their own corresponding login information can log into their accounts.
Database	The database stores all the data for the system (grades, exam times, exam bookings, exam schedules, etc) so that the website can call the information when it needs to be viewed.
Website	The website connects the users to the applications and allows them to perform several operations (booking exams, scheduling exams, etc). The website must have a clean and usable ui so that navigation within the website is simple and intuitive.

The above diagrams Sequence diagram for Use Case 3 and 6. The sequence diagram #1 displays a student booking an exam and sequence diagram #2 displays a lecturer scheduling and exam. Both diagrams have the incorrect password loop as this is a feature that both lecturers and students will need to sign in with their own respective accounts. The scheduling requires a loop for the scheduling of the exams as its required if we don't get the time slot we wanted for the exam.

Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design

Not Addressed	**Partially Addressed**	**Completely Addressed**	**Design Decisions Made During the Iteration**
		UC3	Module required for the exam scheduling system to work correctly was implemented
		UC6	Modules of the preliminary interfaces and base structure on the exam system were implemented
	CRN-1		No relevant decisions made.
	CRN-2		No relevant decisions made.
	CRN-3		No relevant decisions made.