Faculty of Engineering & Applied Science

SOFE 3650U
Software Design & Architecture

Deliverable 3

Group#: 18

Section CRN#:43509

Due Date: November 12th, 2021

First Name	Last Name	Student Number
Saro	Karimi	100755079
Tirth	Patel	100751761
Vatsal	Patel	100728903
Zain	Butt	100751676

Project title: University of Canada Course Registration (UCCR)

Deliverable 3 – Design of the Use Case (Due Dec. 6) This portion of the project should follow the ADD design process. The format of the submission should mimic the FCAPS case example in the textbook[1]. All 3 iterations of the design process are expected and all 7 steps should be included.

Review Inputs

<table>
<tr><th>Category</th><th>Details</th></tr>
<tr><td>Design purpose</td><td>To produce design for an early estimation and define structure of the system.</td></tr>
<tr><td>Primary functional requirements</td><td>Primary functional requirements is represented by the following use cases:
<table><tr><th>Primary Use Case</th></tr><tr><td>UC-4: Course Registration</td></tr><tr><td>UC-5: Course Drop</td></tr><tr><td>UC-6: View Course Schedule</td></tr><tr><td>UC-9: Modify Course</td></tr></table></td></tr>
<tr><td>Quality attribute scenarios</td><td><table><tr><th>Scenario ID</th><th>Importance to the customer</th><th>Difficulty of Implementation According to the Architect</th></tr><tr><td>QA-1</td><td>Medium</td><td>High</td></tr><tr><td>QA-2</td><td>High</td><td>High</td></tr><tr><td>QA-3</td><td>Medium</td><td>Medium</td></tr><tr><td>QA-4</td><td>High</td><td>Low</td></tr><tr><td>QA-5</td><td>High</td><td>High</td></tr></table>QA-1,QA-2 and QA-3 are chosen as drivers for this iteration.</td></tr>
<tr><td>Constraints</td><td>All the constraints are selected as drivers with the exception of CON-1,CON-2, CON-4 and CON-7.</td></tr>
<tr><td>Architectural concerns</td><td>All architectural concerns are selected as drivers.</td></tr>
</table>

Iteration 1: Establishing an Overall System Structure

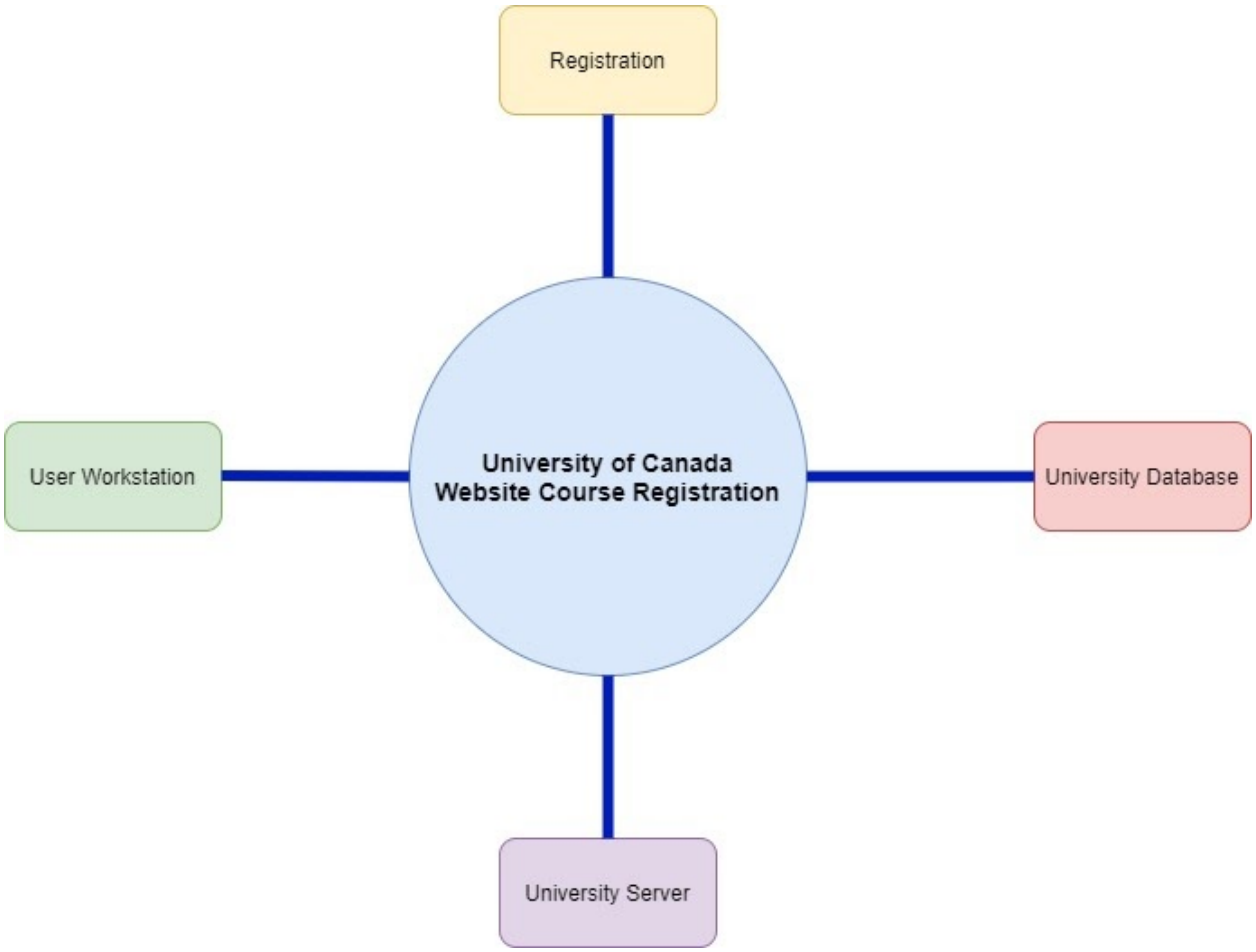
Step 2: Establishing Iteration Goal by Selecting Drivers

The goal of this iteration is to create a system structure for course registration as we are using greenfield mature domain for our design. The following table summarizes the important quality attributes and constraints that need to be addressed.

QA-1	Performance
QA-2	Modifiability
QA-3	Availability
QA-5	Usability
CON-3	A backup of all data should be stored for the registration time frame.
CON-5	Lecturers are not allowed to teach more than 3 courses in a semester.
CON-6	Students are allowed to choose 6 courses at max.
CON-8	Students cannot drop their courses after week 7 of starting the classes.
CRN-1	Structuring a server that can handle many requests at the same time.

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

As we discussed in the “Step 2: Establishing Iteration Goal by Selecting Drivers” , the greenfield mature domain design is being used and therefore, the entire course registration system is selected to refine. The figure below shows the system elements that are being refined.



Choose One or More Design Concepts That Satisfy the Selected Drivers

<table>
<tr><th>Design Decision and Location</th><th>Rationale</th></tr>
<tr><td>Structuring the system client portal using the Web Application</td><td>We would like users to interact with our application through a web browser. Rich user interface is not required to fulfill important quality attributes or constraints and requires the application to be installed on the user machine. QA-1,QA-2,QA-3 and CON-3 will be achieved by using Web Application Reference Architecture and uses minimum client-side resources.

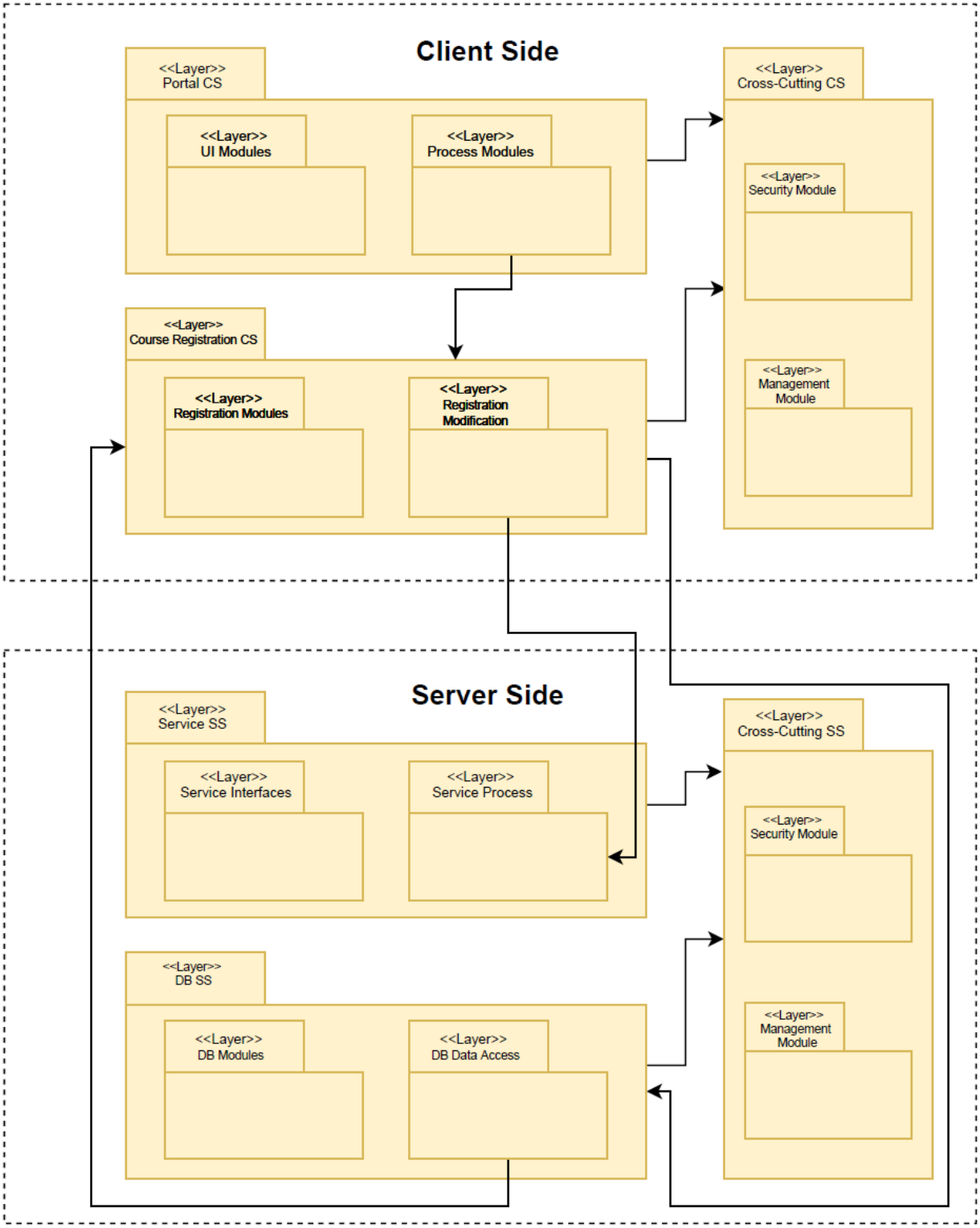
Discarded Alternatives:
<table>
<tr><th>Alternatives</th><th>Reason for Discarding</th></tr>
<tr><td>Mobile Applications</td><td>The idea of using mobile applications is good for many projects but for this one it’s not as good and it’ll be useless to have it. The system shouldn’t be accessed by mobile phones and that's why we discarded this component.</td></tr>
<tr><td>Rich Client Application(RCA)</td><td>Having RCA in a project makes many things easier and it’s probably one of the hardest decisions we made for discarding it. The reason we are not using RCA is retrieving information from the internet and using a two-tier architecture while a web application uses multi-tier architecture. Also, the website must be accessed only by web browser which is not possible in RCA.</td></tr>
</table></td></tr>
<tr><td>Service application on the server side</td><td>Service side is the system that runs on the server. The service agent part from the reference architecture is adjusted to extract the access to the time server. Service applications don’t give a UI yet rather administration that is consumed by different applications. No different choices were thought of and disposed of, as the engineer knew about this reference architecture and thought of it as completely adequate to meet the requirement.</td></tr>
</table>

The **UI** of the registration portal will be created using the bootstrap library which needs a small knowledge in HTML, CSS, and Javascript.	Bootstrap is a front-end framework which allows you to develop a very good looking and easy to use user interface and it doesn't need that much of knowledge to use it. If the developers have a small experience in HTML, CSS, and Javascript, they will be able to create the front-end of the project. This will meet the needs of the user for having an easy working and simple to use UI.
Backend of the project will be using the **LAMP** structure which is a group of open-source softwares	Developers must be capable of using the "LAMP" for creating the server and the backend of the project. TypicallyLAMP is Linux as the operating system, Apache as the Web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. That will be our real challenge for the developers

Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

Design Decision and Location	Rationale
Storing local data in the **Web application** temporary and then move and save it to the **Server Application** Side	CON-3 requires to backup the local data during the registration time frame just in case if the application lost connectivity.
Allowing changes by some specific clients to make changes in the **Server Application** from **Client Side**	Like the admin and UC-12 UC-8 UC-9 UC-10 which relates to QA-2.

Sketch Views and Record Design Decisions



Element	Responsibility
Portal Client Side (CS)	This layer is responsible for user interaction to the application.
User Interface (UI) Modules	The point of client-machine contact and communication in a device is the user interface (UI). Display displays, keyboards, mouse, and the look of a desktop are all examples of this.
Process Modules	Process modules offer users with a logical mapping between their purpose and the application's function.
Course Registration CS	This layer consists of modules that will allow users to interact with course registration on the client side.
Registration Modules	These modules implement operations to present the registration process. For instance, displaying course data.
Registration Modification	Any modifications to the registration modules are performed here.
Security Modules	These modules are responsible for protecting the information stored in the system.
Management Module	These modules dictate the management services that are offered by the system.

Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During the Iteration
		UC-4	Choosed the reference architecture which establishes the modules that will support this functionality. The functionality is used by specific users.
	UC-5		Choosed the reference architecture which establishes the modules that will support this functionality.
	UC-6		Choosed the reference architecture which establishes the modules that will support this functionality.
	UC-9		Choosed the reference architecture which establishes the modules that will support this functionality.
		UC-12	Choosed the reference architecture which establishes the modules that will support this functionality.
QA-1			No relevant decisions were made during this iteration in which it was not important to address.
		QA-2	Identification of the elements for the first step of this quality attribute have been addressed but it should not be more investigated in the next iterations.
	QA-3		Identification of the elements for the first step of this quality attribute have been addressed but it should be more described in the next iteration.
		CON-3	By using web application and server application methods we are able to store everything in the database in case of any data loss or outage so that we can reuse that information after the problem is solved.
CON-5			No relevant decisions made.
	CON-6		By using the bootstrap library, the process of sending an error message and processing the data entered by the user will be easier. But relevant decisions have not been made yet.

	CON-8		Choosed the reference architecture that contains modules will help support this functionality.
		CRN-1	A chosen reference architecture and deployment pattern.
	CRN-2		Choosed a reference architecture that satisfies this functionality partially.

Iteration 2: Identifying Structures to Support Primary Functionality

Step 2: Establish Iteration Goal by Selecting Drivers

The goal of this iteration is establishing a way to address the connection between some important Quality Attributes with the use-cases and if possible explain a way to implement the Constraints in the use-cases. Our focus is on QA-1 and QA-2. We consider the UC-4, UC-5, UC-9, and UC-12. This is important because this is the first step for the development team to make the system.

The primary use cases are listed below as follows:

- UC-4: Course Registration
- UC-5: Course Drop
- UC-9: Modify Course
- UC-12: Profile Updating

The drivers are listed below as follows:

- QA-1: Performance
- QA-2: Modifiability
- CON-3: A backup of all data should be stored for the registration time frame.

Choose One or More Elements of the System to Refine

Modules located in the Course Registration CS layer will be refined further. Since, the primary functionality of the system is course registration, refining course registration modules will refine primary functionality.

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

Design Decision and Location	**Rationale and Assumptions**
The use of Three-Tier Architecture Development	The web application needs to be able to access the database to store/retrieve . Three-Tier Architecture is a great choice as it runs on its own infrastructure and can be updated or scaled as needed without impacting the other tiers. Furthermore, it provides faster development, improved scalability, and better security.

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