
Course Plan

Cégep de la Gaspésie et des Îles
Montreal Campus

Continuing Education Service

Mobile Application Development
LEA.C8

End of Program Project **420-ENV-MT**

WEIGHTING: 0-20-3 DURATION: 300 hours UNITS: 7,67

PROFESSORS:

Lucas Bran
Poshtareh Pargol
Richard GrandMaison
Sakkaravarthi Ramanathan
Zied Zaier, PhD.

Contact : Omnivox MIO
LBranch@cegepgim.ca
PPoshtareh@cegepgim.ca
RGrandMaison@cegepgim.ca
SRamanathan@cegepgim.ca
zzaier@cegepgim.ca

Fall 2018
Group MAD 304

1. GENERAL COURSE DESCRIPTION

The student is called upon to undergo, by means of a practical project, a work-related experience related to the profession of mobile application development. More particularly, the student will familiarize himself or herself with the practice of the profession and apply most of the competencies acquired during the academic program.

2. PLACE OF THE COURSE WITHIN THE PROGRAM

The end of program project is offered in the final semester.

3. COMPETENCY AND ELEMENTS OF THE COMPETENCY

- ENG1: Integrate into the workplace

Elements of the competency	Performance criteria
1. Become familiar with the reality of the profession.	1.1 Complete collection of information on the practical organization of the business. 1.2 Attentive observation of the work context. 1.3 Active participation in different tasks of the profession.
2. Integrate knowledge, skills, attitudes and skills acquired during the academic program.	2.1 Correct establishment of connections between interventions in the workplace and the knowledge acquired during the training.
3. Be aware of changes in perception acquired during the internship	3.1 Establishment of profitable exchanges with members of the team. 3.2 Correct evaluation of the consequences of the project on the perception of the profession and the choices of a job.
4. Produce an internship report.	4.1 Thorough description of the interventions realized during the project. 4.2 Nuanced discussion of the project's contribution to professional development. 4.3 Correct determination of the difficulties encountered and suggestions for improvement.

4. CONTENT

1. INTRODUCTION

This chapter introduces the client and the projects. It defines the terms of reference and also gives a brief overview of the remaining chapters to guide the reader.

1.1 The Client

Established in 2016, the Gaspésie Systems Group (GSG) is both the longest standing and largest system development group in the Department of Computer Science at the *Cégep de la Gaspésie et des Îles*. GSG conducts basic and advanced development into all aspects of software systems extending from basic information systems to mobile applications. They currently focus on four key overlapping topics: software engineering, Web application, mobile computing, and middleware.

1.2 Projects' Background

The Gaspésie Systems Group is trying to implement different mobile application based projects. A summary of the objectives and goals of each one of the projects is available below.

Project 1. The Gaspésie Systems Group is creating an app for scheduling appointments with Pedagogical Counsellors (PC) in the Cegep. This app will be accessible to all PCs who wish to request a meeting with a student, provide his availabilities during a specific day (Start time and end time) and will be able to share this information on the app and to all students who wish to schedule an appointment with a PC or accept a meeting to participate. Each user will be able to create a profile and have their record kept track of for appointments, meetings and remind events based on your profile. This app can be functional on either iOS or Android.

Project 2. Develop an app that enables the users to store all the health records - may it be their medical reports, medicine prescriptions, family doctors' contact numbers, blood test reports, etc. - in a web based services. The users can access their records as well as share them with doctors and physicians for assisting them in providing better medical solutions. The best part is that a user can keep and maintain the record of his/her whole family via a single app.

Project 3. The Gaspésie Systems Group is creating an app to help students find lunch buddies. The app will allow a user to pick a day, start time and end time for which they want to find a lunch buddy. They can specify if they want to eat at the CEGEP or outside. If they eat outside, they can specify parameters such as budget or cuisine type. They can specify if they want their post to be public to all other users, or only to a subset of users. They can also accept others request for a lunch a buddy. They can access a history of their posted/accepted lunch buddies. They can view a list of their upcoming lunches. Works on iOS or Android.

Project 4. The Gaspésie Systems Group is creating an app to help students share and find best daily deals. This mobile application provides access to offers of activities, travel, goods and services. This app will allow students to discover and save on of great deals at nearby restaurants, things to do, shopping, travel and more. This mobile application will act not only as a deals searching platform but also as a deals sharing tool between students. This app can be functional on either iOS or Android.

1.3 Document Summary

Chapter 2 is a summary of the objectives and goals of the projects

Chapter 3 raises some issues that need to be considered before starting the projects. It also addresses the methodology of the projects' management.

Chapter 4 looks at the definition of the four (4) parts of the projects as well as the responsibilities and deliverables of each team that will be working on each project.

Chapter 5 deals with some open issues in the area of the projects' individual and team final evaluation that need further consideration.

2. Project Description

Gaspésie Systems Group needs a product for each one of the projects, described in section 1.2, which consists of frontend layer, service layer and core backend. The future products will be used on a few student *Cégep de la Gaspésie et des Îles* sites as part of the product roadmap is to document and engineer the user interface for end users (the Cégep students). Each project should focus on a mobile application – which is currently not implemented.

Each project, described in section 1.2, should focus on the design, implementation and user experience of the mobile application, but also consider technological feasibility and business viability, i.e. the services are important to the *Cégep de la Gaspésie et des Îles* sites.

Each project should also focus on the service layer to answer all frontend (mobile application) information requests and core backend to retrieve, store and manage the system information.

There is no limitation put on the technology applied – input and creativity from young, enlightened and inspired students is what we are after.

3. Project Management Approach

In this section we discuss the team organization and responsibilities and describe the project documentation, deliverables and timelines.

3.1 Team Organization

We need to have a team (4 to 6 students) for each project (refer to section 1.2) to address the different parts of the project (refer to section 4). Lots of different skill sets are required for a successful team. The team members' composition must be submitted at the beginning of the project and approved by the teachers. Team projects will be approved only at the discretion of the teachers. **Each student** must contribute to the project (from home or CEGEP), **attend project meetings** (in person or over video conferencing software), and **participate in the team final presentation** (in person). Teachers can rearrange the teams based on the amount of students, the skills of the students, and mentors (teachers supervising the teams) we have available.

3.2 Project Responsibilities

While the specific roles and responsibilities for the project team members will be assigned at the beginning of the project, all team members are responsible for the final delivery of the project. The team mentors (teachers supervising the teams) are a subset of the project team and are responsible for the project management and leadership activities such as initiating, planning, executing, monitoring, controlling, and closing the various project phases.

3.3 Project Documentation Organization

Project documents will be kept primarily on the team documentation server. This server is shared between team members and the mentors. The team documentation server will be accessible through your Web browser with an encrypted username and password protection. Each document is added to the team documentation server under a revision number and placed in appropriate folders. The team documentation server will use an efficient document control software to avoid document overwriting in the case of similar revisions being added to the server at the same time.

3.4 Project Timeline

The project time line reflects all tasks necessary to complete the project as well as the estimated time required for each project part. *Cégep de la Gaspésie et des Îles* is assuming 300 working hours per team member. This assumption is reflected on the project schedule ending on February 22, 2019.

3.5 Project Deliverables

Teams shall be accountable for the following deliverables:

- System design documentation.
- Screen design and flow.
- Database design document.
- Database deployment script
- Mobile application source code
- Backend application source code

4. Work Description

Each one of the projects, described in section 1.2, consists of frontend layer, service layer and core backend. We need to have a team for each one of the projects to address these different parts on the project. As a team member, you may focus mainly on the client-side, on the server-side, or both. This is related to what you want to build based on the following parts of the project. What is important is what you build rather than which side you focus on. Some of the following project parts may require an in-depth knowledge in JAVA programming, and Mobile Programming alone. Others may require Database and documentation knowledge. Still others may use a good mix of both.

4.1 Project Documentation

This project part consists of teamwork with 2 deliverables, namely a Project Analysis Report and the database schema script of your project. Your deliverables will be evaluated on completeness as well as level of thought, and attention to principles discussed in class during your MAD program.

Create your Project Analysis Report using “Microsoft Word” and prepare your ORACLE database schema script using a simple text file with “SQL extension”. Only hardcopy submissions of your Project Analysis Report will be accepted. A hardcopy of your database schema script must be included in your final report. Please refer to Appendix A for more details. It is obligatory to use for your Project Analysis Report the template available in Appendix A of this document; submissions non-compliant with this template will not be accepted.

As described in Appendix A, this part also consists of 2 goals namely the project UX (user experience) and project UI (user interface). Your team will have to research pre-existing solutions to the problem and determine the best solution in regard to the problem the client requested. This part contains the user end-point experience meaning your team will have to figure out who will be the user of the software and how they will use it. The solution determined would

then be showcased to the client where he will determine if any changes would be required. Wireframe software must be used such as JUSTINMIND or Balsamiq. An interactive prototype must be deployed for the client. Your team must save all iterations within Github to be submitted at the end. Iteration may consist of new features or changes the client requested. When the client approves the final solution proposed by your team you may begin the front end implementation and print a hardcopy of the wireframe to be submitted.

4.2 Project Backend

This project part consists of teamwork with 3 deliverables, namely the project database interface, project JSON building and parsing and the web service calls and responses of your project. Your deliverables will be evaluated on completeness as well as level of thought, and attention to principles discussed in class during the MAD program. Create your Project Code using “Netbeans IDE” or “Eclipse IDE” (only one single Integrated Development Environment – IDE – per team) and prepare your code commits using a simple text file with “SQL extension”. Only a GitHub submissions of your Project will be accepted. A documentation of your code must be included in you final GitHub submission. Please refer to Appendix B for more details. It is obligatory to use for your work the requirements available in Appendix B of this document; submissions non-compliant with these requirements will not be accepted.

4.3 Project Frontend

This part consists of teamwork with 3 deliverables, namely the project code, the project UI(user interface) and project front-end testing. Your team will be tasked to create the iOS/Android application from the wireframe and the clients overall request. The UI must match the wireframe and if a roadblock does happen, your team must create a new iteration of the wireframe and explain to the client. Your team will be tasked to create the code required to make all the features requested workable. The last phase will consist of your team testing the application. The application must be tested manually and internally by code. The test code and the manual user test must be documented and a hard copy must be printed at the end to be handed in to the client. All of the Project Frontend must be recorded within Github with all minute changes.

Assessment table:

<p>Four (4) Deliverables</p>	<ol style="list-style-type: none"> 1. Become familiar with the reality of the profession. 2. Integrate knowledge, skills, attitudes and skills acquired during the academic program. 3. Be aware of changes in perception acquired during the internship 4. Produce an internship report. 	<p>40%</p>	<p>To be determined</p>
<p>Term-end evaluation: Application source code</p>	<ol style="list-style-type: none"> 1. Become familiar with the reality of the profession. 2. Integrate knowledge, skills, attitudes and skills acquired during the academic program. 3. Be aware of changes in perception acquired during the internship 4. Produce an internship report. 	<p>25%</p>	<p>To be determined</p>
<p>Term-end evaluation: Project Report</p>	<ol style="list-style-type: none"> 1. Become familiar with the reality of the profession. 2. Integrate knowledge, skills, attitudes and skills acquired during the academic program. 3. Be aware of changes in perception acquired during the internship 4. Produce an internship report. 	<p>25%</p>	<p>To be determined</p>
<p>Term-end evaluation: Project Presentation</p>	<ol style="list-style-type: none"> 1. Become familiar with the reality of the profession. 2. Integrate knowledge, skills, attitudes and skills acquired during the academic program. 3. Be aware of changes in perception acquired during the internship 4. Produce an internship report. 	<p>10%</p>	<p>To be determined</p>

The teams of teachers will perform, 4 weeks after the project start date, a mid-term project performance feedback of all teams.

If any team receives a mark of zero (0) due to late submission of work or other reason(s) the team will receive a written warning from both the team's supervising teachers and from the academic adviser.

(Any team which does not respect any additional deadline(s) risks to fail the end of program project. This means that each team member could fail with a final mark under 60%)

Appendix A

Project Report Template

The following annotated template shall be used to complete the Software Requirements Specification (SRS) part of your project. The instructor must approve any modifications to the overall structure of this document.

1. Introduction

The introduction to the Software Requirement Specification (SRS) document of your project should provide an overview of the complete SRS document. While writing this document please remember that this document should contain all of the information needed by a software engineer to adequately design and implement the software product described by the requirements listed in this document

1.1 Purpose

What is the purpose of this mobile application?

1.2 Scope

This subsection should:

- (1) Identify the mobile application to be produced by name; for example, Host DBMS, Report Generator, etc*
- (2) Explain what the mobile application will, and, if necessary, will not do*
- (3) Describe the application of the mobile application being specified. As a portion of this, it should:
 - (a) Describe all relevant benefits, objectives, and goals as precisely as possible. For example, to say that one goal is to provide effective reporting capabilities is not as good as saying parameter-driven, user-definable reports with a 2 h turnaround and on-line entry of user parameters.*
 - (b) Be consistent with similar statements in higher-level specifications (for example, the System Requirement Specification), if they exist. What is the scope of this mobile application product?**

2. General Description

This section of the SRS should describe the general factors that affect the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

2.1 Product Functions

This subsection of the SRS should provide a summary of the functions that the mobile application will perform.

2.2 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements.

3. Specific Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the requirements that are used to guide the project's software design, implementation, and testing. Each requirement in this section should be:

- *Correct*
- *Traceable (both forward and backward to prior/future artifacts)*
- *Unambiguous*
- *Verifiable (i.e., testable)*
- *Prioritized (with respect to importance and/or stability)*
- *Complete*
- *Consistent*
- *Uniquely identifiable (usually via numbering like 3.4.5.6)*
- *Follow MOSCOW approach.*

Attention should be paid to carefully organize the requirements presented in this section so that they may be easily accessed and understood. Furthermore, this SRS is not the software

design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

3.1 Functional Requirements

This section describes specific features of the mobile application project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transactions shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc). Non-Functional Requirements are related to:

- *Performance*
- *Reliability*
- *Availability*
- *Security*
- *Maintainability*
- *Portability.*

3.3 Other Requirements

Catchall section for any additional requirements.

4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

4.1 Use Case Diagrams

Create a use-case diagram for your mobile application. Your design will be evaluated on completeness as well as level of thought, and attention to principles discussed in class. Your design should also provide good horizontal distribution of your project's functionality and allow for all described features to be developed, as much as possible.

4.2 Sequence Diagrams

Provide a sequence diagram for 3 major use cases. Your design will be evaluated on completeness as well as level of thought, and attention to principles discussed in class. Your design should also provide good horizontal distribution of your project's functionality and allow for all described features to be developed, as much as possible.

4.3 Class Diagrams

Your class diagram should display all major classes in the system, each class's attributes (fields), relationships between classes, and named directed associational relationships with multiplicity between classes. Your design will be evaluated on completeness as well as level of thought, and attention to principles discussed in class. Your design should also provide good horizontal distribution of your project's functionality and allow for all described features to be developed, as much as possible.

4.4 Screen flows diagram and Screen design

It's fairly common the need to transition between multiple App Screens, we will explore in this section the way to manage those transitions. In this first part of this section, we will show how the screen flows are defined using State diagram to drive each screen transition as shown in figure 1.

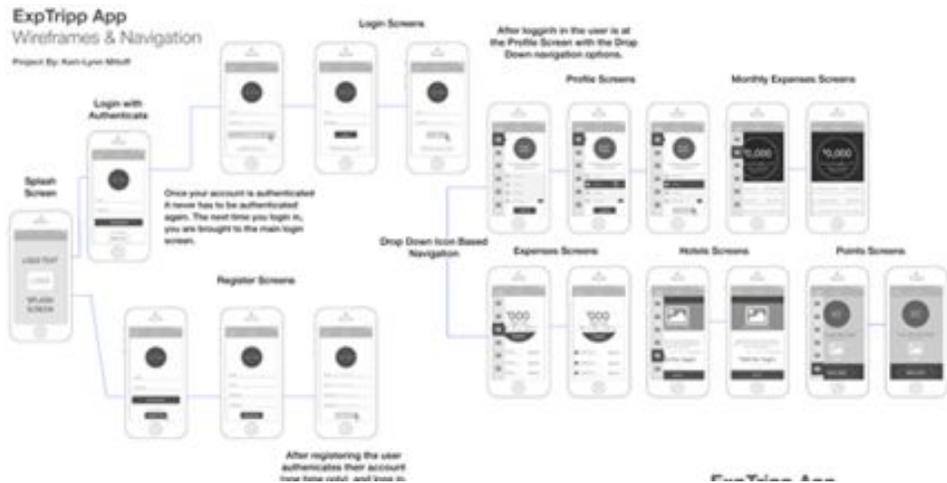


Figure 1. Screen flows diagram

Good navigation, like good design, is invisible. Applications with good navigation just feel intuitive and make it easy to accomplish any task. While there may be many options for navigating content within an app, we want to focus on screen layouts for primary and secondary navigation and interactive design principles. In this second part of this section, we will show how the main 2 screens off your mobile application are designed, as shown in figure 2.

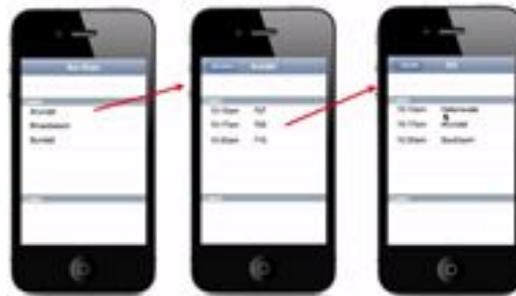


Figure 2. Screen design

5. Database

5.1 Database Schema

Your Database Schema should display all major database tables in the system, each table's attributes (fields), relationships between tables, and named primary keys with foreign keys between tables.

5.2 Database Script

Your Database Script should allow the creation of all major database tables in the system, each table's attributes (fields), relationships between tables, and named primary keys with foreign keys between tables.

6. Web Services' calls details

Create a Web Services' calls description, as per the following tables, for your mobile application. Your description will be evaluated on completeness as well as level of thought, and attention to principles discussed in class. Your design should also provide good horizontal distribution of your project's functionality and allow for all described features and screens to be developed, as much as possible.

URL	http://www.localhost:8080/myJoke/main/userProfile&zzaier&12345678
Call	UserProfile
Parameters	username = zzaier Password = 12345678
Response	Example 1 { "Satus": "OK", "Timestamp": 1478351819, "UserName": "zzaier", "Fname": "Zied", "Lname": "Zaier", "PostalCode": "H3S1k1", "Phone": 51455555555, "Active": true } Example 2 { "Satus": "OK",

	<pre> “Timestamp”: 1478351819, "UserName": "zzaier", "Fname": "Zied", "Lname": "Zaier", “PostalCode”: "H3S1k1", “Active”: true } Example 3 { “Satus”: "WRONG", “Timestamp”: 1478351819, "UserName": "zzaier" } </pre>		
Keys	“Satus”	Status of the response Possible values : “OK”, “WRONG”, “ERROR”	Mandatory
	“Timestamp”	UTC format response timestamp	Mandatory
	“UserName”	The user name of the user	Mandatory
	“Fname”	The first name of the user	Optional
	“Lname”	The last name of the user	Optional
	“PostalCode”	The postal code of the user	Optional
	“Phone”	The phone of the user	Optional
	“Active”	The status of the user profile Possible values : true, false	Optional

URL	http://www.localhost:8080/myJoke/main/townsList&1478351819
Call	townsList
Parameters	Last time the mobile App received the information = 1478351819 (the value 0 if the first use of the app)
Response	Example 1 <pre> {“Satus”: “OK”, “Timestamp”: 1478351900, "Townns": [{ " id": 1000, </pre>

```

    "name": "Montreal",
    "ModifiedTimestamp": "1478351900",
    "Active": true
  },
  {
    "id": 1001,
    "name": "Quebec",
    "ModifiedTimestamp": "1478351911",
    "Active": true
  },
  {
    "id": 1002,
    "name": "Toronto",
    "ModifiedTimestamp": "1478351922",
    "Active": true
  }
]

```

```
}

```

Example 2

```

{
  "Satus": "OK",
  "Timestamp": 1478351819
}

```

```
}

```

Example 3

```

{
  "Satus": "ERROR",
  "Timestamp": 1478351819
}

```

Keys	"Satus"	Status of the response Possible values : "OK", "ERROR"	Mandatory
	"Timestamp"	UTC format response timestamp	Mandatory
	"Towns"	The list of towns changed since the last call	Optional
	"id"	The id of the town	Optional
	"name"	The name of the town	Optional
	"ModifiedTime Stamp"	UTC format town modification timestamp	Optional
	"Active"	The status of the town Possible values : true, false	Optional

Appendix B
Backend Project Requirements

Workflow based web service composition. The project will deal with modeling the workflow of particular scenarios and dynamically composing web services to satisfy the workflow. Existing standards can be used to model workflows for instance RESTful service. A sample scenario can be a typical adding a new book to sale or list existent books for sale. Emphasis will be on how the web services are designed, and how they can be composed at runtime based on the requirements. This project will give students good exposure to service oriented architecture.

Workflow based database composition. The project will deal with modeling the workflow of particular scenarios and dynamically implementing the database to satisfy the workflow. Existing Database Managing System (DBMS) can be used to model workflows for instance ORACLE DBMS. A sample scenario can be a typical storing information of a new book to sale or retrieve information about existent books for sale. Emphasis will be on how the database is designed, and how it can be interrogated at runtime based on the requirements. This project will give students good exposure to database implementation and architecture.

1. Attendance in class and for practical training

1.1. Regular participation in the course (activities inside and outside the classroom) is considered essential for skills mastery.

1.2. Class attendance will be recorded at each class.

1.3. Students arriving late for class may be refused access by the teacher.

1.4. It is the student's responsibility to provide the teacher or academic advisor with a reason for an absence. During a practical activity, the student must give a reason for an absence to the activity sponsor and to the supervisor

1.5. For continuing education purposes, the following reasons can be given to justify an absence: legal situations (supported by a document), death of a family member, health care (with a professional's note) and parental obligation.

1.6. However, even in the case of justified absences, the student is at risk of being expelled from the course after a maximum of 20%. The teacher will inform the coordinator, who will decide what action to take.

1.7. A student who must be absent for an undetermined period because of exceptional circumstances must inform the academic advisor, who will decide what action to take.

1.8. Presence at practical activities is compulsory. An absent student may be expelled from his or her practical activity for an unjustified absence. The activity's supervisor will inform the academic advisor, who will decide what action to take.

1.9. A student absent without a reason deemed valid by the teacher must not expect the teacher to provide special attention aimed at making up missed learning.

2. Language quality

2.1. In accordance with the Institutional Language Policy, the Continuing Education Service must strive to ensure students' language quality.

2.2. For assignments and exams, the summative assessment of language quality counts for 10% of the mark.

2.3. In programs and courses where mastery of the written language is a learning objective, the Continuing Education Service reserves the possibility of setting the summative assessment of language quality at more than 10%. The decision will be made by the program team.

2.4. The Continuing Education Service reserves the right to use the clauses concerning language quality that are found in the Departmental Learning Policy of the departments in which continuing education courses and programs originate. Thus, the maximum credit given for language may, in some cases, exceed 10%.

Clause 2.9.3 of the Institutional Policy on the Evaluation of Student Achievement

2.9.3 For allophone clients and immersion students, 10% of marks will be given for language quality in the first year, but a student will be allowed to make up entirely the 10% if he or she makes corrections to an assignment within 5 days. In the second year, a total of 10% will also be given for language quality. A student can make up half (5%) if he or she makes corrections to an assignment. In the third year, the same rules apply for all students.

For exams taken in class, there is no penalty for language quality for allophone students.

3. Presentation of assigned work

3.1. The manner in which assigned work must be handed in will be in accordance with the usual methodological standards. Students should consult the student guide delivered to them during the orientation activity.

3.2. The deadlines set for handing in written work and presenting activities must be met.

3.3. The student is responsible for handing in his or her work before the deadline, which must be respected even if the student is absent.

3.4. A penalty of 10% will be applied for each late day up to a maximum of 50% (5 days) to work handed in late. The student must give the teacher prior notice that work will be handed in late; if not, the work can be refused. After 5 days, unless there are extenuating circumstances, a mark of zero will be given for work handed in late.

4. Pass Mark

Clause 2.4 of the Institutional Policy on the Evaluation of Student Achievement

2.4.1 The pass mark is 60% (Section 27, College Education Regulations)

5. Cheating and plagiarism

Clause 2.12 of the Institutional Policy on the Evaluation of Student Achievement

2.12.1 All cheating, attempt to cheat or collaboration in cheating will result in a mark of zero for the test or work involved. In such cases, the teacher must seize the documents and make a report,

which must be sent to the centre's office by the department coordinator or academic advisor for continuing education. The use of MP3 players, cell phones and laptop computers (unless authorized) is prohibited.

2.12.2 A student who plagiarises, that is, who steals or passes off as his or her own any work whatsoever, regardless of the source, of any author without giving a proper credit to the author will be given a mark of zero for the work handed in

2.12.3 Any subsequent cheating or plagiarism by that student will result in a mark of zero for the entire course involved.

6. Review of marks

Clause 2.15 of the Institutional Policy on the Evaluation of Student Achievement

2.15.1 Any student who wants a review of the mark given for assigned work or for a summative assessment test during a session must ask the teacher within 5 working days following receipt of the mark.

Review of the mark for a final assessment test

2.15.2 Any student who wants a review of the mark given for a final assessment test, must address his or her request to the academic office or the academic advisor for continuing education within 10 working days following the entry of the final mark in Omnivox.

2.15.3 The department must form a review committee made up of three teaching staff members (including the teacher concerned). The committee will send the final mark after review to the Studies Department. For continuing education, the review committee must be made up of two teaching staff members and include the teacher concerned as well as the pedagogical counsellor.

September 2017