SEG 3101 PROJECT - DELIVERABLE 1

Submitted
to
Wassim El Ahmar

by:

Group 22
Sami Hassan 3001691285
Uyen Nguyen, 300173303
Yi Yau Wong, 300241845
Tiana Ye, 300189296
Kristen Duong, 300240425

October 13, 2023
University of Ottawa

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Purpose of Project Document

This deliverable aims to present the user requirements, entailing both functional and non-functional ones, that we have created for the Graduate Admission Management System (GAMS). These requirements have been revised so that there are no duplicates. They are also sorted by priority in the table using a scale that follows Low Priority (L), Medium Priority (M) and High Priority (H) scaling. The methods used for filtering and prioritizing the requirements are explained in the Discussion section.

1.2 Vision of Project

GAMS aims to enhance the initial steps of graduate studies for students. It enables seamless communication between graduates and related personnel to ensure the success of students. By guiding users step-by-step, GAMS simplifies complex operations into easy and achievable procedures. The efficient design of this program will boost the university's competitiveness and ensure that potential students receive their offer ahead of all other universities. GAMS guarantees the accomplishment of harmony between various departments of the admission team, students and those interested in the University of Ottawa.

2.0 USER REQUIREMENTS

Requirement Number	Requirement	Type	Priority Low Priority (L) Medium Priority(M) High Priority (H)
1	The system shall allocate the applications to the corresponding academic advisor based on the program.	FR	H
2	The system shall allow the academic assistant to perform calculations for the admission average of students.	FR	H
3	The system must be able to accept or refuse applications based on a student's average.	FR	H

4	The system must update an application’s status when a decision has been made.	FR	H
5	The system must notify the student when a supervisor has been assigned to them.	FR	M
6	The system must generate and send an email containing details of eligible students, that are without a supervisor, to all professors in their program	FR	H
7	The system should automatically admit a student that is admissible to a course/project-based program.	FR	H
8	The system should produce a recommendation of admission for those admitted to a course/project-based program.	FR	H
9	The system must allow Student Services to produce and send a set of documents to students confirming their acceptance/admission.	FR	H
10	The system shall automatically reject an offer if the student does not accept it within ten working days.	FR	H
11	The system shall make an offer invalid after one semester since the issue date.	FR	H
12	The system must be able to digitize any document uploaded in PDF format.	FR	M
13	The system shall automatically save any application drafts.	FR	M
14	The system should be able to generate an evaluation template with a mail merge involving data from an Excel file (produced by copying/pasting from uoCampus) and Word macros.	FR	H
15	The system must divide all student applications evenly among members of the committee.	FR	L

16	The system should notify students when their applications' status changes.	FR	M
17	The system shall send an email receipt to a student once their payment has been processed.	FR	H
18	The system must refuse any application with an average below a CGPA 7.0.	NFR	H
19	The system must only take the average grade of the last 20 courses, if for master's, or the last 10 courses, if for a doctorate.	NFR	H
20	The system shall be available in English.	NFR	H
21	The system shall be available in French.	NFR	H
22	The system should have a different user interface for different programs.	NFR	L
23	The system must authenticate users when they log in (security purposes)	NFR	H
24	The system shall provide distinct interfaces for course/project-based and thesis-based programs.	NFR	M
25	The system shall offer varying degrees of automation and integrated systems for acceptance.	NFR	L
26	The system must load all data within 1.5 seconds.	NFR	H
27	The system must have a search bar with filters to find an applicant in the database based on identifiers.	NFR	H
28	The system must limit the number of applications received from a student to 20.	NFR	M
29	For any uploaded data, the system must encrypt it (security purposes).	NFR	H

3.0 DISCUSSION

3.1 Filtering/cleaning Strategy and its Success

A filtering/cleaning strategy used was by addressing any vague or ambiguous requirements and adding specificity and clarity to it. Ambiguities often lead to misunderstandings; therefore, by addressing them, the system requirements would be concise and clear. This ensures that there is no miscommunication between the parties involved, such as between stakeholders and developers. We are trying to define what requirements should be part of the project and what requirements are out of the scope.

Another strategy was by taking any requirements that were too similar, and combining them. This helps to prevent redundancy and repetition for developers who can now focus on a single, well-defined requirement instead of duplicating work. With that being said, a system without repetition ensures that the development process is efficient and the system is not costly to develop.

When considering the system requirements, it is important to ensure that they are relevant to the system's objectives and goals. This will lead to a project's success and its ability to stay true to its intended purpose. Requirements must also be flexible and accommodate for potential changes in the world (sustainability). As technology evolves and new user needs are discovered, these requirements must be future-proof in the case that the system would ever need a major rework. If these requirements can easily adapt to changes and be sustainable, a system overhaul will not be required and resources (time, money, skills, etc) can be spent improving and/or innovating new fields.

3.2 Prioritization Strategy

After filtering the requirements, the next step is ranking the requirements into high, medium, and low priority. The purpose of ranking the requirements is to distinguish between what is important and what is optional.

If a requirement is listed as a high priority, that requirement is essential to the main goal of the system, which is to process graduates' applications to their graduate study programs. Whereas medium priority means the requirement is beneficial to the system, if met, but is not necessary for the system to function. Low priority refers to requirements that can be developed if there is extra time or labour.

When using high, medium and low levels to determine the prioritization of requirements, this allows for a simple and quick understanding of a system, instead of extensive explanation and documentation. To add to that, each priority level is distinct, making it clear which requirements hold high importance and need to be addressed as a priority.

Another prioritization strategy involved creating functional requirements first and then non-functional afterwards. The reason for this is that functional requirements are more important to the workflow of the whole program. Therefore, defining the functional requirements first supported how the non-functional requirements were to be generated.

4.0 REFERENCES

No external references were used to complete this deliverable.