|  |  |  |
| --- | --- | --- |
| Students at a graduation ceremony | | Books |
| Graduation RequirementsSimplifier | | |
| ICS370Fall 2020Bradley Taylor, Rene Ntumnui, David Qual |  | |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Table of Contents**  Introduction …………………………………………………………………………….3  Business Case ………………………………………………………………………...4  Feasibility Study ………………………………………………………………………5 -7   * Section 1 Technical Feasibility ………………………………………5 * Section 2 Financial Feasibility ……………………………………….6 * Section 3 Economic Feasibility ………………………………………6 * Section 4 Political Feasibility …………………………………………7 * Section 5 Market Feasibility …………………………………………..7   Use Cases ………………………………………………………………………………..8 – 10   * Actor/Use Case Table ……………………………………………………8 * Student Use Cases ………………………………………………………..9 * Administrator Use Cases ……………………………………………….9 * Use Case Diagram …………………………………………………………9 * Estimated Cost ……………………………………………………………..10 * Hardware and Software Requirements ………………………….10 | | |
| Introduction Our vision includes development of an application that allows current students enrolled at Metropolitan State University to easily track their progress toward graduation. This lightweight application is a solution that will cover all major courses of study at Metro State. It will allow any student to quickly view the required courses for their major program of study as well as any technical and general electives. | | |
|  |  |
|  | |
| Business Case The average bachelor’s degree requires 120 college credits. However, the average recipient completes 134 credits before graduating. There are several factors that contribute to this phenomenon. We decided to explore those factors and determine what can be done to help with the problem.  One reason that students take extra classes is because they fail to take advantage of advising. Most, if not all universities, offer extensive academic advising services. However, advisors can be overloaded on cases and utilizing their services can take a great deal of time and effort. Thus, many students fail to follow through.  Other factors involve changing majors and/or transferring to different schools. Requirements can change from school to school and from program to program. This can cause great confusion and ultimately contributes to students taking unnecessary classes.  Degree audits can be very difficult to read. There are categories laid out in the audits, but often there are conditions listed that become very complicated. The wording of these conditions can almost resemble legalese, making them hard to interpret.  In addition to the factors mentioned above, I recently sat through student introductions for a college course. While giving their introductions, four separate students stated that they did not know when they would complete their major. It was apparent that they did not grasp the requirements to graduate with a degree.  Our desire is to create a simple application that will allow a student to better understand the requirements to graduate a given course of study. This system will have a simple user interface that can be accessed over the internet. There will be no appointments to make with advisors and no complicated degree audit to read. The scope will be programs at Metropolitan State University. | | |

#### Feasibility Study

**Introduction**

This project feasibility report has been prepared to support our development of a software solution intended to be used by students that attend Metropolitan State University. We intend to show that this project will be beneficial across a variety of spectrums. These spectrums include, technical, financial, economic, and political. We will also explore the potential market value of this project.

**Section 1 – Technical Feasibility**

We considered the technical feasibility of the project from two different viewpoints. This allowed for us to get a full picture of whether the project could be completely funded from start to finish.

The first was project development and infrastructure. After researching several tools and integrated development environments, we found that we could utilize freeware in some instances and low-cost subscription SaaS products in others.

The second viewpoint required us to consider the technology of all potential end users. We determined that all of the students that attend Metropolitan State have access to laptop or desktop computers. Many have their own devices, in other circumstances, the University has computer workstations available for student use.

In summary, each of the products that will be needed by developers are supported by the current technical resources that we have on hand, no additional hardware will need to be purchased. Also, there will be no costs associated with end users, which is ideal. This will allow us to stay within our allotted budget for development and infrastructure, a topic that will be discussed in more detail in the following section.

**Section 2 – Financial Feasibility**

Software development projects often involve numerous technical and operational expenses, particularly in the early phases. For this undertaking, we have gone to great lengths to avoid that potential pitfall.

Our project staffing consists of three individuals working together as a team. As mentioned in the previous section, all of the needed hardware for the project is in hand.

We do expect to incur minimal expenses related to software, this may come in the form of licensing or subscription fees related to development and hosting of the solution.

**Section 3 – Economic Feasibility**

The previous section explored the costs associated with developing this solution, this section will focus on the ability of the solution to break even or become profitable in the future.

We do anticipate a considerable incubation period for this software feature before it becomes widely used and appreciated. This is largely due to factors beyond our control. The University follows yearly cycles of enrollment, registration, and course completion and evaluation. These cycles repeat throughout Fall, Spring, and Summer semesters. Graduation requirements are not always considered daily but tend to coincide with the aforementioned semester cycles. This phenomenon will contribute to the incubation period.

The next section of this report will explore additional factors that may contribute to a potentially lengthy break-even period. We understand that this is not always ideal for a software project. However, the nature of our project will afford us the ability to allow several semesters to elapse as students and advisors grow in their appreciation of this powerful tool.

**Section 4 – Political Feasibility**

In a previous section, we mentioned that there could be a considerable period of time that elapses before this system becomes widely used. Our study on political feasibility helped us to form that belief. We have, however, concluded that none of the factors considered were harmfully significant or insurmountable.

Our project will be designed with the intention of helping students and also advisors. We expect the latter to involve political implications. The reason being that any outside attempt to improve processes or procedures often implies that a weakness or deficiency currently exists.

Previous experience shows that attempts to improve processes or procedures within any group or department, often incurs initial backlash from stakeholders or actors. We anticipate that, in this case, the benefits will be significant enough that the any pushback will be short lived.

**Section 5 – Market Feasibility**

Our completed offering will have great potential within the Metropolitan State University system. We envision that students will find the solution to be intuitive and easy to use. It will negate the need to schedule countless appointments with their advisor and will provide peace of mind.

Advisors will also greatly benefit from this solution. Informed students will free up advisors to perform their essential duties and help them to manage their caseloads more efficiently.

Finally, we expect this solution to encounter demand from other colleges and universities. This simple application will improve student experience and advisor efficiency and thus will be coveted throughout the industry. This factor greatly mitigates the lengthy break-even period previously discussed.

#### Use Cases

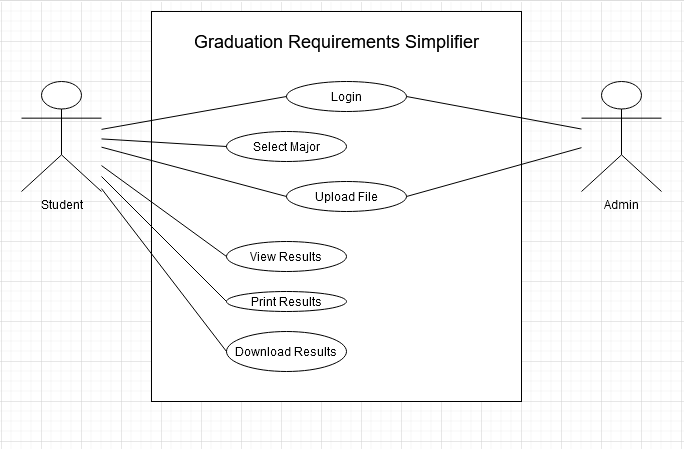
|  |  |
| --- | --- |
| Users/Actors | Use Cases/Functionalities |
| Student | * Login * Select Major * Upload Transcript/Dars Report * View Results * Print Results * Download Results |
| Administrator | * Login * Upload CSV File |

**Student**

The student will be the primary actor in our system. The student will have to be able to login to access the interface of the program. Upon gaining access to the interface, the student will need access to several program features. These features include the ability to select a major course of study, upload a transcript or Dars Report, run the program, and access the results. Additionally, the student will need to view, download, or print the results.

**Administrator**

There will be a system administrator that needs to interact with the system. The administrator will need to be able to login to access the administrative features of the program. Upon gaining access to the administrator interface, the instructor will need the ability to configure and maintain program majors and requirements.



#### Estimated Cost

With the average cost of a developer being around 60,000 a year cost to develop this would likely be 200 hours or so. So cost from a development perspective would be around 5800 dollars. You could leave it at that but it would be best to include an analyst that could double as a quality assurance because they would focus on documentation and breaking your program so users do not. That would be roughly equal to the developer so another 5800 to 11600 if you wanted to separate the roles. With this being a web product you have options to host it yourself or let another company host it. AWS is a fantastic option for hosting and for a single instance with 100 gigabytes of SSD storage, four core cpu, 16 gigabytes of ram is about 185 a month. So total cost to develop, test is between 11600 and 17400, with a monthly cost of 185 dollars for AWS hosting. This could reduced by utilizing more frugal means, but this is the cost that we came up with based on current costs.

#### Hardware and Software Requirements

**For development**

Realistically web development can be done with nearly any device with a text editor so this greatly reduces the need for any high spec device or development software. There is also a need for a database manager. We use MySQL as a database so the system would need to be able to use that. The requirements are rather low, so an i3, 4gb of ram and 128gb hard drive (preferably an SSD) are the minimum requirements. You would also need to install the MySQL software.

**For users**

This would be a website with the only need being a modern browser to facilitate access to most websites. The website would be accessible by mobile devices but the best experience would be from a computer.