**Software Requirements Specification**

**For**

**SRCCMSTHS – EGS**

**Version 1.0 Approved**

**Prepared by Joshua C. Dimapilis**

**October 9, 2013**

**Table of Contents**

**Table of Contents**

**Revision History**

**1. Introduction......................................................................................................................................1**

1.1 Purpose.................................................................................................................................................. 1

1.2 Document Conventions..................................................................................................................... 1

1.3 Intended Audience and Reading Suggestions............................................................................. 1

1.4 Product Scope ..................................................................................................................................... 1

1.5 References...............................................................................................................................................1

**2. Overall Description..........................................................................................................................2**

2.1 Product Perspective ............................................................................................................................ 2

2.2 Product Functions................................................................................................................................. 2

2.3 User Classes and Characteristics....................................................................................................... 2

2.4 Operating Environment....................................................................................................................... 2

2.5 Design and Implementation Constraints......................................................................................... 2

2.6 User Documentation ........................................................................................................................... 2

2.7 Assumptions and Dependencies ...................................................................................................... 3

**3. External Interface Requirements ..................................................................................................3**

3.1 User Interfaces...................................................................................................................................... 3

3.2 Hardware Interfaces.............................................................................................................................. 3

3.3 Software Interfaces................................................................................................................................ 3

3.4 Communications Interfaces ............................................................................................................... 3

**4. System Features................................................................................................................................4**

4.1 System Feature 1................................................................................................................................... 4

4.2 System Feature 2 (and so on)............................................................................................................ 4

**5. Other Nonfunctional Requirements.............................................................................................4**

5.1 Performance Requirements................................................................................................................ 4

5.2 Safety Requirements............................................................................................................................. 5

5.3 Security Requirements.......................................................................................................................... 5

5.4 Software Quality Attributes.................................................................................................................. 5

5.5 Business Rules ........................................................................................................................................ 5

**6. Other Requirements ........................................................................................................................5**

**Appendix A: Glossary............................................................................................................................5**

**Appendix B: Analysis Models..............................................................................................................5**

**Appendix C: To Be Determined List..................................................................................................6**

1. **Introduction**
   1. **Purpose**

**The purpose** of this requirements specification is to help its readers obtain a glimpse of the software at hand.

The report is duly made to help de-obscure gray areas within the software made.

* 1. **Document Conventions**

There were no specific standards to how the document was made, but the requirements that are listed in the report are all in sequential order according to priority.

* 1. **Intended Audience and Reading Suggestions**

The following Specifications are advisable to be read by the **Faculty and Students** of Senator Renato "Compañero" Cayetano Memorial Science and Technology High School. For the clients to have a clear glimpse of the software at hand, the entire section is to be read from beginning to end.

* 1. **Product Scope**

The authors plan on assisting the school in providing a user-friendly grading system that will benefit both the students and the teachers. If time will permit, we plan on doing the following:   
  
1. A Viable System for Enrolment, Registration and Grading   
2. A Database of the Students and their vital information for main purposes in CRUD, (Create, Read, Update and Delete)   
3. An application for accessing grades for the students   
  
In line with the plan mentioned above, the authors plan on creating user – friendly interfaces also, (similar to the cited reference) to help the target audience in using the proposed system.

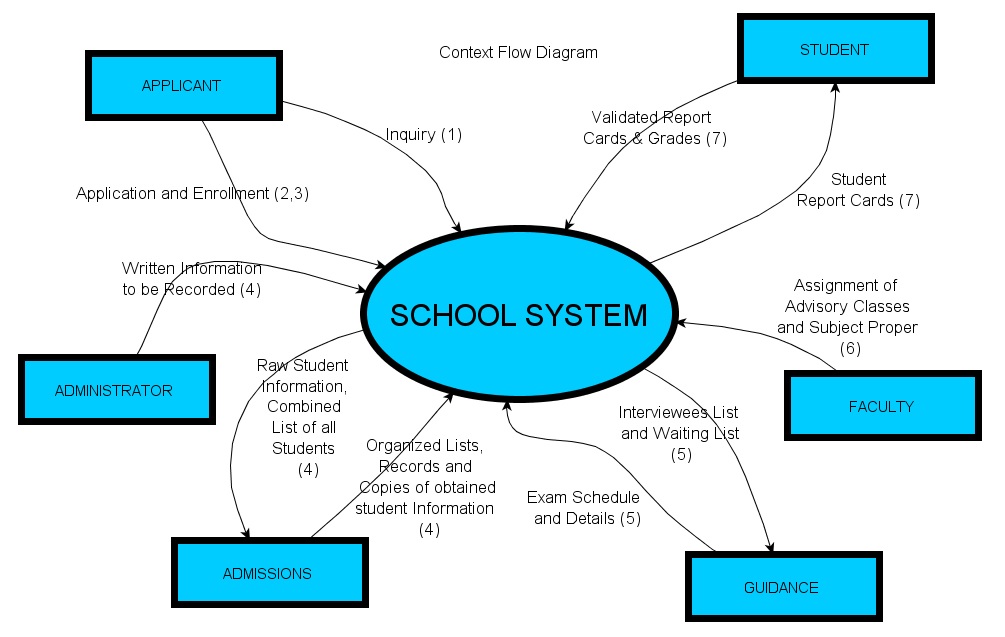
* 1. **References**

**Wiki:**

[http://projects2.apc.edu.ph/wiki/index.php/Senator\_Renato\_"Compañero"\_Cayetano\_Memorial\_Science\_and\_Technology\_High\_School\_-\_School\_System](http://projects2.apc.edu.ph/wiki/index.php/Senator_Renato_%22Compa%C3%B1ero%22_Cayetano_Memorial_Science_and_Technology_High_School_-_School_System)

1. **Overall Description**
   1. **Product Perspective**

The product being done by the team, is not from a specific line of software, but a replacement towards the manual way of encoding and data recording. Past data which were recorded through the use of Spreadsheets and the like, are now to be changed to gear towards data repositories and databases hosted online or through a local server. The application is not genuinely taken from a larger composite structure, instead it is the structure itself that is being referred to in the Specification. Below is a Context Flow Diagram that illustrates the structure and its involvement with the keyplayers.



* 1. **Product Functions**

The main functions of the product, in accordance to project type, are as follows:

1. **Enrollment**
   1. **Applicant Recording (Creating, Reading, Updating and Deleting)**
   2. **Test Scheduling (Creating, Reading, Updating and Deleting)**
   3. **Student Listing (Segregating and Displaying)**
2. **Grading** 
   1. **Student Recording (Creating, Reading, Updating and Deleting)**
   2. **Aggregate Grade Computing (Creating, Reading Updating and Deleting)**
   3. **Grade Listing (Computing, Displaying)**
   4. **User Classes and Characteristics**

The following user classes will most likely be utilizing the software:

1. Applicants – upon enrollment, they will use this upon entering basic information about themselves, \*given only if administration allows applicants to make use of their computers
2. Students – the students may frequently use the system to check on their grades during the quarter
3. Faculty – the faculty will mainly use this, since they will be responsible for encoding first-hand data that are necessary for aggregate function computing
4. Administrators – the admin will most likely be responsible for the handling of the users that will be in the system, it is also part of the admin’s responsibility to regulate the access given to the software
   1. **Operating Environment**

For a brief description of the intended plan of the authors, the project should be done in line with the topology of the School itself. To implement the proposed solution of the authors, the need for computers must be satisfied, and networks must be built.

* 1. **Design and Implementation Constraints**

As mentioned above, the following constraints have been set for the project:

1. A Viable System for Enrolment, Registration and Grading   
2. A Database of the Students and their vital information for main purposes in CRUD, (Create, Read, Update and Delete)   
3. An application for accessing grades for the students

* 1. **User Documentation**

The following documentation has been done by the researchers in the past:

1. **Documentation for Traditional Systems Development Life Cycle**
2. **Diagrams**
3. **Status Reports**
   1. **Assumptions and Dependencies**

During deployment, the following circumstances are assumed:

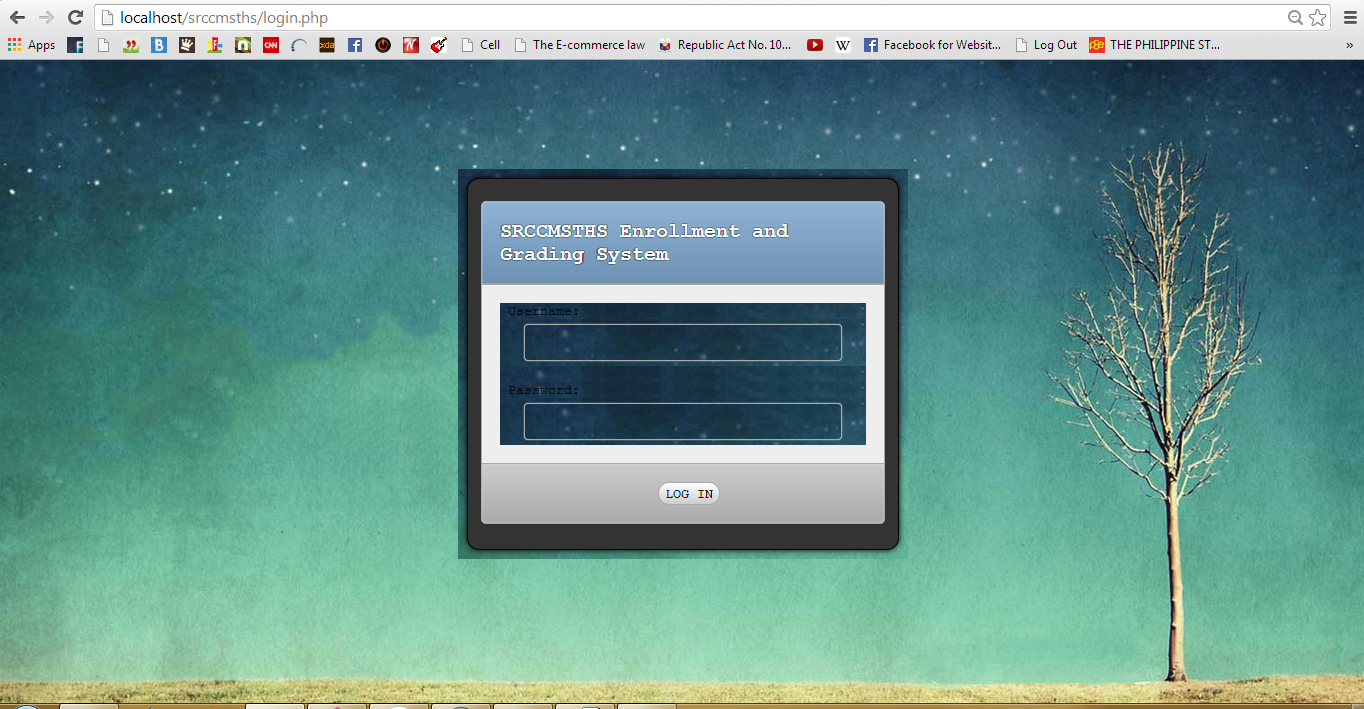
* Users are computer literate individuals
* Computer systems already have present repositories in which stored data are present
* Facilities have viable hardware that can accommodate the necessary requirements of the project

Basically, the entire project is resource – dependent. Time constraints, human resources and the like are also similar issues, however the resources present in the deployment phase of the project are most likely to be assessed to usher the success of the implementation.

1. **External Interface Requirements**
   1. **User Interfaces**

For the user interface, the pattern of the software is similarly made to that of flavio, however some css and html codes may be tweaked towards a more different / minimalistic view that is desired to be achieved.

The picture below is merely a sample homepage, the font – style, the colors and the background are still to be changed.



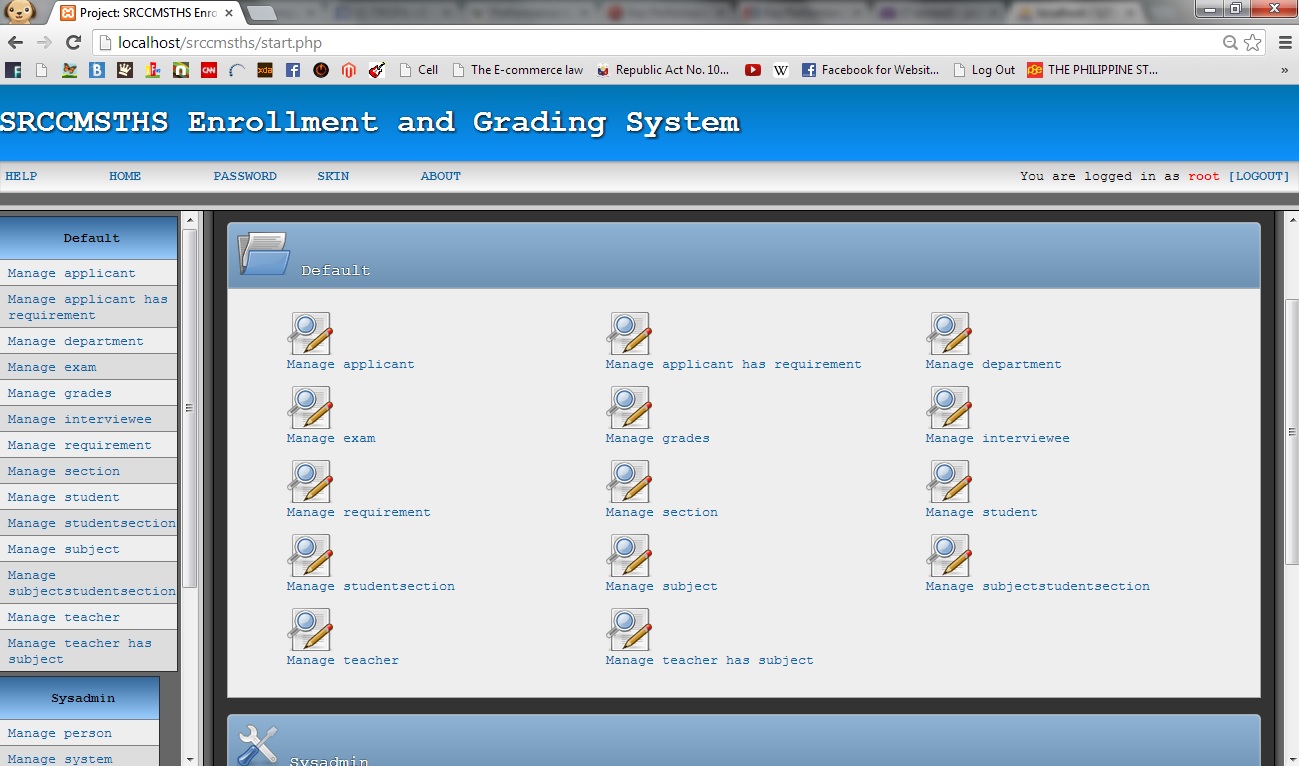
* 1. **Hardware Interfaces**

As a gist, the network design must conform to the star topology because it must be assured that the solution is also economy – wise and does not require much time and effort to be put into action. Unfortunately, matters such as these may only be considered after the assessment of the entire project itself before the prototyping and implementation phase begins.

* 1. **Software Interfaces**

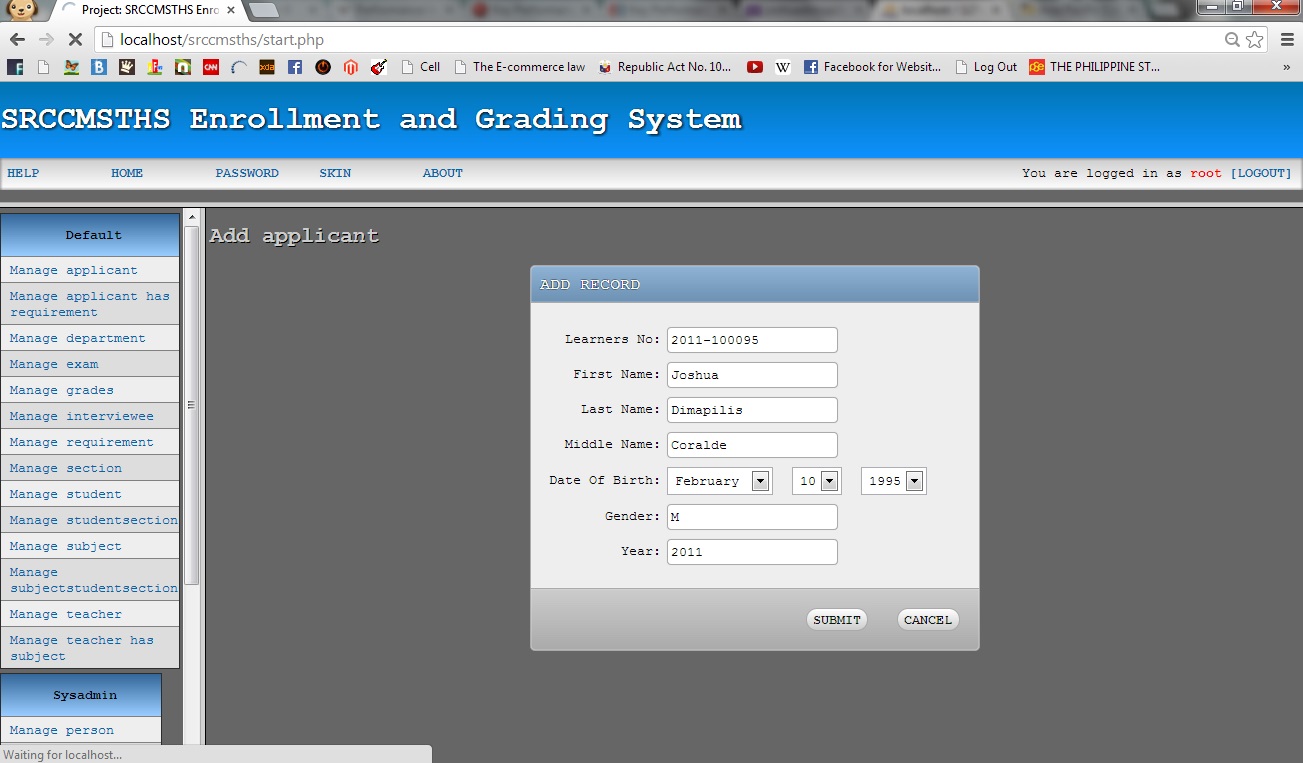
Similarly, as we all do have an idea that there are always two sides of an interface, the software interfaces deal mostly with how the software looks like from the inside. Given administrator privileges, how will the interface be presented to the admin or to the user itself? These questions can be answered mainly by experiencing the use of the software itself.

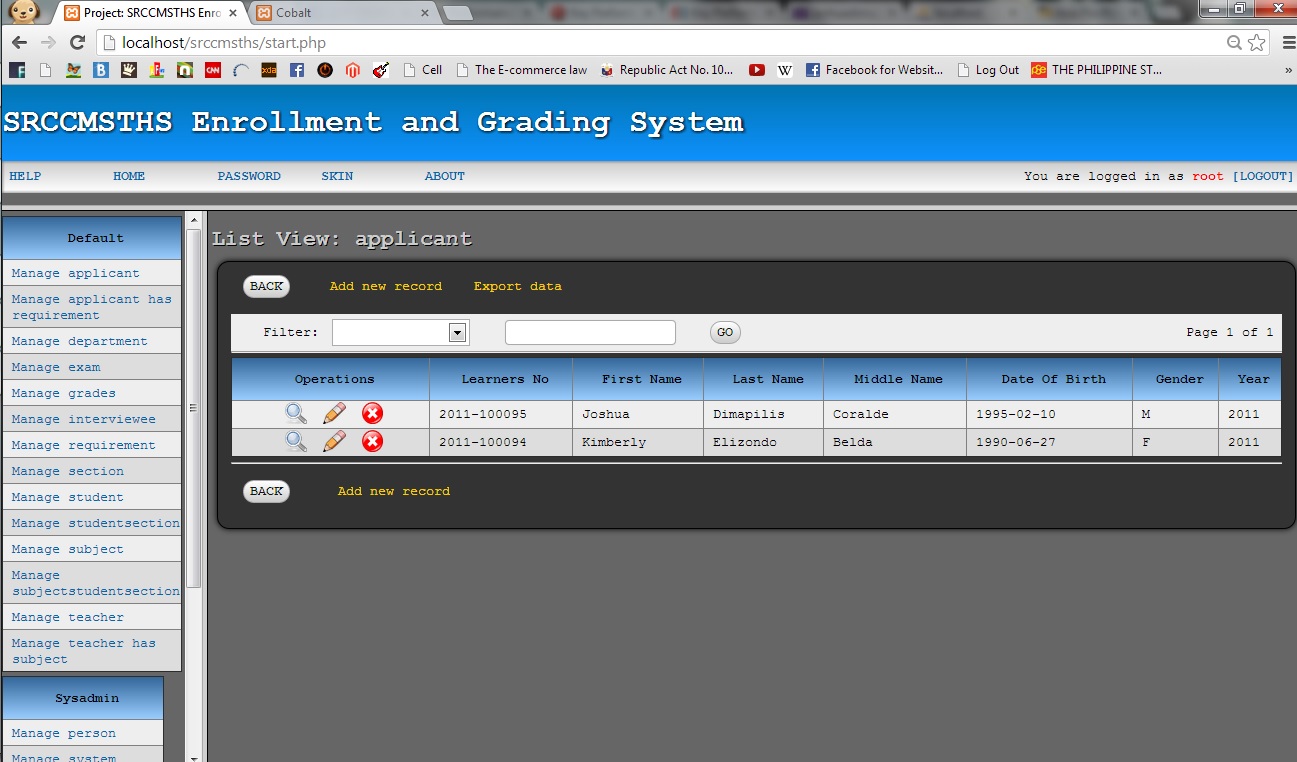
The image below is a representation of the DEBUG User / Administrator Mode of the project.



* 1. **Communication Interfaces**

During the processes encountered by the system users, the communication interfaces are similar to most of the processes involved in the DEBUG User Mode. Other examples are the images that follow.





1. **System Features**

Most certainly, the features of a system are easily defined by its objectives, and by the reasons why they were made. In the following lines, some diagrams are used to illustrate the different functionalities of the proposed system.

* 1. **Creating Objects**

Summarizing all the creation of files in one feature, one basic functionality of the System is creating students, teachers, interviewees, applicants and other objects of the like.

* + 1. **Description and Priority**

There is a High Priority for this functionality, mainly because it’s the standard component of how the users can handle the software. The trade – off is that along with creation, comes the other *crud (create, read, update and delete) functionalities.*

* + 1. **Stimulus/Response Sequences**

The stimulus is the need to input a student or any object that needs to be recorded, and the corresponding response would be the encoding of the student or any object for different matters together with its details.

* + 1. **Functional Requirements**

\*errors to be confronted are to be inserted here\*

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1: ***TBD***

REQ-2: ***TBD***

* 1. **Deleting Objects**

Summarizing all the deletion of files in one feature, another basic functionality of the proposed software is removing students, teachers, interviewees, applicants and other objects of the like which are no longer necessary for the system’s processes.

* + 1. **Description and Priority**

There is a High Priority for this functionality, mainly because it’s a very basic component of how the users can manage the system. Deleting outdated objects / details are necessary for the success of a system.

* + 1. **Stimulus/Response Sequences**

The stimulus is the need to delete a mistake or any object that is no longer needed in the system and the corresponding response would be the removal of the mistaken detail or any object under a different context.

* + 1. **Functional Requirements**

\*errors to be confronted are to be inserted here\*

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1: ***TBD***

REQ-2: ***TBD***

* 1. **Aggregate Functional Computation**

Summarizing all the deletion of files in one feature, another basic functionality of the proposed software is removing students, teachers, interviewees, applicants and other objects of the like which are no longer necessary for the system’s processes.

* + 1. **Description and Priority**

There is a High Priority for this functionality, mainly because it’s a very basic component of how the users can manage the system. Deleting outdated objects / details are necessary for the success of a system.

* + 1. **Stimulus/Response Sequences**

The stimulus is the need to delete a mistake or any object that is no longer needed in the system and the corresponding response would be the removal of the mistaken detail or any object under a different context.

* + 1. **Functional Requirements**

\*errors to be confronted are to be inserted here\*

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1: ***TBD***

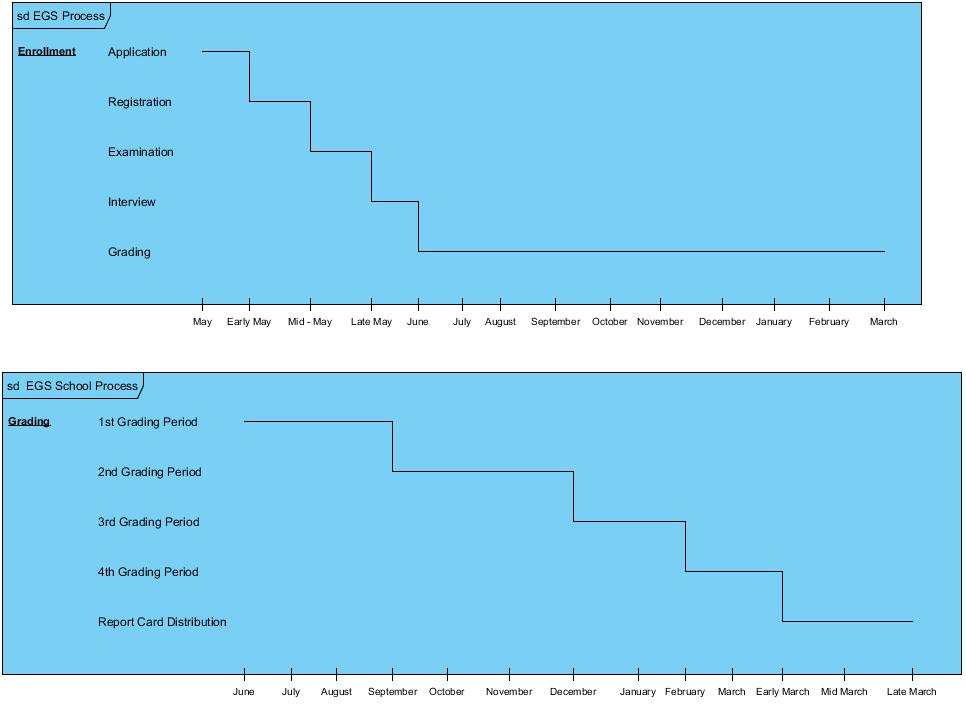
REQ-2: ***TBD***

1. **Other Non-functional Requirements**

There are plenty of other non – functional requirements that are not mentioned in the SRS, however most of the discussion of such are already discussed in the documentation presented by the proponents.

* 1. **Performance Requirements**

Part of the requirements for adequate performances for the system to function is its capability of adapting to the time that is allotted to it. It must be able to answer the needs of the clients during certain times. Below is a diagram that illustrates the different time requirements that must be met during implementation phase.



* 1. **Safety Requirements**

TBD

* 1. **Security Requirements**

TBD

* 1. **Software Quality Attributes**

TBD

* 1. **Business Rules**

TBD

**6. Other Requirements**

Some other requirements are present

**Appendix A: Glossary** TBD

**Appendix B: Analysis Models** TBD

**Appendix C: To Be Determined List**

\*Most functional requirements may be discovered as the software is being finished

\*Safety and security requirements all get answered in the implementation phase

\*\*Quality attributes and Business Rules may be answered / provided by the clients

\*\*\*Appendices to follow