**Chapter 4**

**RESULTS AND DISCUSSION**

This chapter deals with the results of the project implementation to the beneficiary, including the feasibility studies, and the technical description of the internals of the system.

**Requirements Documentation**

The Requirements Documentation shows the feasibility studies, flowcharts for the (manual system), and the Data Flow Diagram.

**Feasibility Studies.** To determine whether the proposed system is feasible for the beneficiary, different data gathering methods were done by the proponents. Those were the following:

1. **Economic Feasibility.** An economic feasibility study was done to determine the costs and benefits of the software and, if the beneficiary can shoulder the costs of the software’s development and maintenance. The costs and benefits in this study falls into two (2) categories, namely:

* Qualitative Cost – Hardware/software, operational, maintenance, and training costs.
* Quantitative Cost – Employee discontentment due to their fear of change which is contrary to the traditional processes that they are comfortable with.
* Qualitative Benefits – Reduction of costs in terms of personnel and operation, as well as increased productivity due to the efficiency of the system.
* Quantitative Benefits – improvements in top level management’s decision making due to the efficiency of the system

Table 4.0

**Software Cost Benefit Analysis**

|  |  |  |
| --- | --- | --- |
| **Type** | **Costs** | **Benefits** |
| **Qualitative** | 1. Reluctance of some teachers to use the system because of fear from using the technology itself | 1. Faster preparation, and execution of the election activity |
| **Quantitative** | 1. Monthly subscription to the domain/hosting company if the beneficiary wants to use a premium add-on (database etc.), custom domain, or add extra storage in the server | 1. Reduction in costs of operation, personnel cost, and possibility of increased revenue if the beneficiary decides to add more functionalities to the website. |

The Table 4.0 shows the potential costs and benefits of the proposed system.

1. **Technological Feasibility.** This feasibility study deals with the capabilities of the beneficiary’s hardware (i.e. computers) to implement the proposed system.

Table 4.1

**Hardware and Software Comparison**

**(Beneficiary vs. Recommended Specs)**

|  |  |  |
| --- | --- | --- |
|  | **Beneficiary** | **Recommended**  **(Minimum)** |
| **Operating System** | Windows 7 32-bit | Windows 7 32-bit |
| **Browser** | Google Chrome | Internet Explorer 9 |
| **Processor** | Intel Core2duo E8400 @ 3.0Ghz | Intel Core2duo E8400 @ 3.0Ghz |
| **RAM** | 2 GB DDR3 @1666Mhz | 1GB DDR3 @1666Mhz |

Based on the Table 4.1, the beneficiary meets all the minimum requirements of the proposed system for it to run smoothly.

1. **Schedule Feasibility.**  This feasibility study shows if the software can be done in a very efficient manner in a given period. If the software development takes longer than it should, then the system is deemed not feasible in this aspect.

Table 4.2

**WEB-BASED VOTING SYSTEM**

**FOR STUDENT GOVERNMENT ELECTIONS**

**OF MATER DEI ACADEMY - GANTT CHART**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | Task Name | Duration | Start | Finish | Recourse Name |
| 1 | **PLANNING** | 61days | 7/13/2018 | 9/14/2018 |  |
| 2 | Form Project Team | 1day | 7/13/2018 | 7/13/2018 | Prof Jayson R. Hermogenes |
| 3 | Make Project Proposal | 1day | 7/13/2018 | 7/20/2018 | Eduria, Torres, Reyes, Garrido |
| 4 | Approved Project Proposal | 1day | 7/20/2018 | 7/20/2018 | Eduria, Torres, Reyes, Garrido |
| 5 | Conduct Data Gathering | 14days | 7/20/2018 | 8/16/2018 | Eduria, Torres, Reyes, Garrido |
| 6 | Observation | 1day | 8/2/2018 | 8/22/2018 | Eduria, Torres, Reyes, Garrido |
| 7 | Interviews | 1day | 8/22/2018 | 8/272018 | Eduria, Torres, Reyes, Garrido |
| 8 | Questionnaire/Surveys | 1day | 8/27/2018 | 9/7/2018 | Eduria, Torres, Reyes, Garrido |
| 9 | Organize/Summarize/Review Requirements | 3days | 9/9/2018 | 9/12/2018 | Eduria, Bryan Joel M.  Garrido, Erica Mae C. |
| 10 | Submission of Requirements | 1day | 9/14/2018 | 9/14/2018 | Eduria, Torres, Reyes, Garrido |
| 11 | **Feasibility Study** | 14days | 9/18/2018 | 10/1/2018 |  |
| 12 | Technical Feasibility | 3days | 9/18/2018 | 9/21/2018 | Eduria, Torres, Reyes, Garrido |
| 13 | Operational Feasibility | 3days | 9/22/2018 | 9/25/2018 | Eduria, Torres, Reyes, Garrido |
| 14 | Economical Feasibility | 3days | 9/26/2018 | 9/29/2018 | Eduria, Torres, Reyes, Garrido |
| 15 | Submission of Feasibility Study Documents | 1day | 10/1/2018 | 10/1/2018 | Eduria, Torres, Reyes, Garrido |
| 16 | **Analysis** | 7days | 10/2/2018 | 10/9/2018 |  |
| 17 | Make Data Flow Diagram | 4days | 10/2/2018 | 10/6/2018 | Eduria, Torres, Reyes, Garrido |
| 18 | System Flowchart | 2days | 10/7/2018 | 10/9/2018 | Eduria, Torres, Reyes, Garrido |
| 19 | Program Flowchart | 2days | 10/7/2018 | 10/9/2018 | Eduria, Torres, Reyes, Garrido |
| 20 | Submission of Analysis Documents | 1day | 11/5/2018 | 11/5/2018 | Eduria, Torres, Reyes, Garrido |
| 21 | **Design** | 10days | 10/15/2018 | 10/20/2018 |  |
| 22 | Data Dictionary | 5days | 10/15/2018 | 10/20/2018 | Eduria, Torres, Reyes, Garrido |
| 23 | Entity Relationship Diagram(ERD) | 3days | 10/15/2018 | 10/18/2018 | Eduria, Torres, Reyes, Garrido |
| 24 | Hierarchical Input-Process-Output | 1day | 10/20/2018 | 10/20/2018 | Eduria, Torres, Reyes, Garrido |
| 25 | Volume Table of Contents (VTOC) | 1day | 10/20/2018 | 10/20/2018 | Eduria, Torres, Reyes, Garrido |
| 26 | Submission of Design Documents | 1day | 11/5/2018 | 11/5/2018 | Eduria, Torres, Reyes, Garrido |

The Table 4.3 shows the duration of time in which the project was undertaken. It took a total of 145 days for the project to be finished.

**Systems Flowchart.** The process of the beneficiary’s existing system is depicted in this part. A flowchart displays the processes of a system in such a way that it will be very easy for end-users to understand.

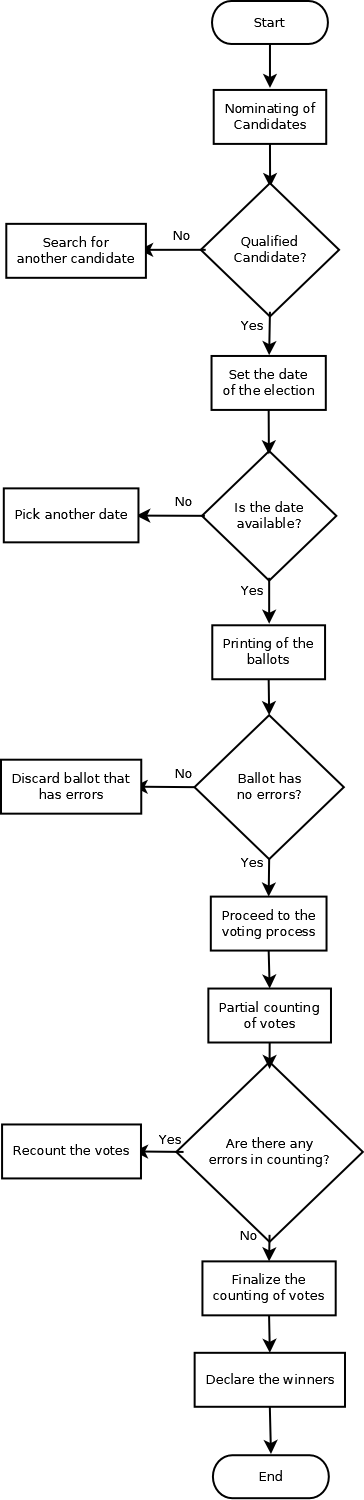


Figure 4.0

**System Flowchart**

**(Existing System)**

**Data Flow Diagrams.** The Data Flow Diagram, or simply DFD, describes the flow of data/information in a system, from the data sources, processes, data storage and eventually, down to the end users. Its aim is to show the overview of a system without going down too much on technical details, which can be done using other diagrams such as the Hierarchical Input-Process-Output (HIPO) diagrams etc. Shown below are the two (2) DFDs created by the proponents for both existing and proposed systems.

**Design of System**

This section contains the different methods that were used to define the internals of the proposed system according to its modular hierarchy, process, and relationship of different objects in the system and the definition of data entities used in the system’s database.

**Hierarchical Input-Process-Output.** The Hierarchical Input Process Output (HIPO) details the hierarchical flow of the input, process and output in every module in the system

**Login Module/Table 1**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | Output |
| Username  Password | Check the validity of the credentials entered before redirecting to the Admin Page | Redirect to the Admin Page |

**School Tab/Table 2**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Add grades  Add section | Take authentication check the validity of the credentials from the database | Redirect to the School tab page |

**User Tab/Table 3**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Add admin  Add user  Import/Export User | Take authentication check the validity of the credentials from the database | Redirect to the Users tab page |

**Election Tab/Table 4**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Add party-list  Add position  Add election | Take authentication check the validity of the credentials from the filled text labels before adding to the database. | Add the credentials to the database. Redirect to the Election tab page. |

**Add Election Tab/Table 4.1**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Edit election  Manage Election | Edit the credentials of the election/Check the validity of the credentials from the database. | Update the database. Redirect to the Add election tab page. |

**Control Numbers Tab/Table 4.2**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Generate Control Numbers  Export Control Numbers | Check validity of the control numbers from the database to generate or export the control numbers. | Redirect to the Control Numbers tab page then show the generated control numbers. |

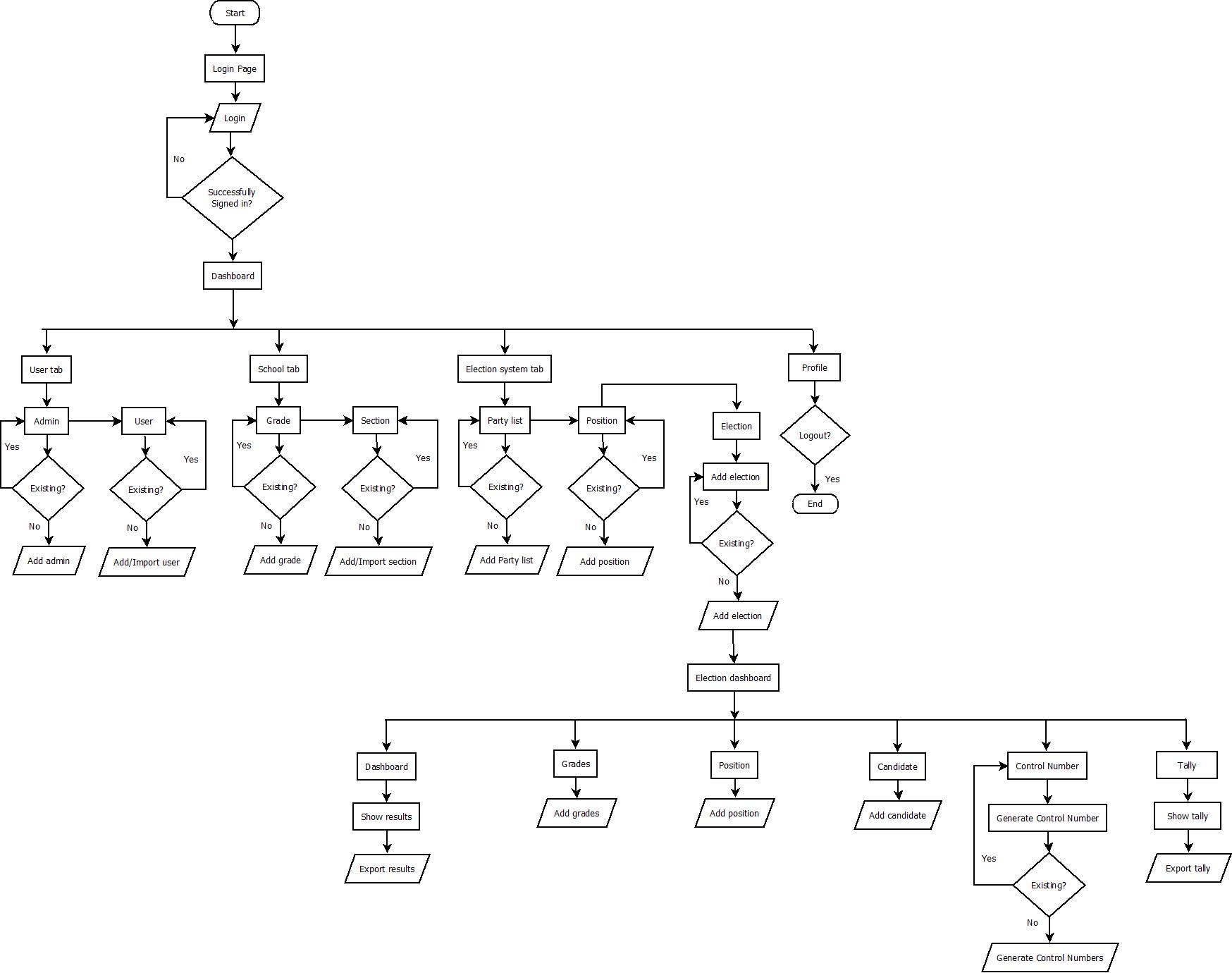
**Manage Election tab/Table 4.3**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Add Candidate  Generate Control Number  Export Control Number | Check the credentials of the Control Numbers and credentials/Check the validity of the credentials from the database | Redirect to the Manage Election tab page |

**Manage Election tab/Table 4.4**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Tally  Declare Winners | Check the validity of the votes in the database | Redirect to the Manage Election Dashboard and Declare Winners |

**Program Flowchart.** For the benefit of the readers, a program flowchart has been created to display the processes of the proposed system in a detailed yet simple and understandable way.



**Entity-Relationship Diagram.**  The Entity Relationship Diagram shows the relationships of the models with each other (which are called “entities”) in the database.

**Data Dictionary.** When a developer is not familiar with the system he is working with, he may find it difficult to figure out the system’s database. That is where the data dictionary is needed. The data dictionary helps the developer understand the database since it describes every variable, including its data types, in the database. This will save much more time for the developer since it will be easier for him to refer to the data dictionary instead of dumping out every variable just to know its contents inside.

**Development and Testing**

After conceptualizing every detail in the system, then comes the actual construction of it. The modules were released in a weekly basis, as based on the principles of Agile Manifesto. But before every release, manual software tests were done to remove every major and obvious bugs which can hamper the functionalities of the system and ensure that it is running smoothly as intended by the beneficiary and the programmers. For more information about the Software Development Methodology that was used in this project, please see the Chapter 3:Technical Background – Methodology.

Figure 6. Project Burndown Chart

|  |
| --- |
|  |

The figure above shows the burndown chart of the project’s progress while it was being developed. The proponents were able to meet the deadline in the development of the software, as shown on the figure.

**Implementation Plan**

A system will never be serviceable if there is no concrete plan that would be enforced for its usage, thus, an implementation plan is needed to ensure the success of the voting system. The proponents devised a set of procedures on how to conduct an election using this web-based voting system that will serve as a guide to the beneficiary for future Student Government Elections.

The implementation plan consists of two (2) phases, namely: the Pre-Implementation and the Operation.

1. **Pre-Implementation**

**1.a. Admin Orientation** – The COMELEC shall undergo training on how to use the Admin and Voter Side of the system. This orientation shall be conducted by the proponents to ensure that the beneficiary will understand the proper usage of the system.

**1.b. Voter Orientation** - After the COMELEC has been briefed about the usage of the system, it is the beneficiary’s duty to provide training among the students on how to use the Voter Side of the system prior to an election event.

1. **Operation**

**2.a. Election Preparation** – The COMELEC must prepare all the data/information that is needed by the system prior to an election event. This includes:

* Basic information of every student (e.g. full name, birthday, address etc.)
* Election Candidates (i.e. Nominees for SG Positions and Party Lists)

**2.b. Creation of Election** – Before the casting of votes can start, an election must first be created by the Admin User. The required fields in the creation of an election must then be filled by the Admin User who is creating the election. The Admin User then shall enter the nominees for each party list and position in the Student Government. After the required fields has been satisfied, the Admin user must generate the control numbers which are to be used by the students as a credential that will enable them to access the Voter Side of the system. The COMELEC must also stress the importance of keeping the confidentiality of the control numbers given to each of the student to guarantee the security of the elections, as someone might use a control number belonging to another student without latter’s knowledge.

**2.c. Voting** ­– This is the part where the voters can now cast their votes on the system, based on the choices provided by the Admin User in the Nomination of Candidates. The proponents are suggesting the use of the plan shown below in the figure on how to conduct the casting of votes in an election, using the proposed system.

Figure 7. Procedures in Casting Votes to the Proposed System

|  |
| --- |
|  |

* The voter goes in the inside the room, but before he/she proceeds to casting votes, a member of COMELEC must check the voter’s identity first, then find his/her name on the list and write the time of his/her entry to the precinct.
* The voter may then cast his/her votes on the computers located inside the precinct. If any of the voters has questions, the COMELEC stationed inside the precinct must always be ready to assist them.
* Once the voter finishes voting, he/she must log out on the list of the voters and provide his/her signature on the space provided.

**2.d. Tallying and Declaration of Winners** – As soon as the casting of votes from the students has finished, the Admin Users can now close the election if it ends earlier than the expected time. If there are ties in the vote results of the candidates, the Admin Users can use the tie-breaker provided in the system which will randomize the results of the tie, providing a winner. Once the tallying has been finished and the list of winners are already finalized, the Admin Users can now declare the winners in the election event.