Digital Voting System

Analysis

# **2**. **Introduction to analysis**

System analysis is the first stage of the software development process in which identify the system problems and their solutions. Analysis is the process of finding a detailed explanation of the system or any organization and understand its nature or to determine its important features (dictionary.com, 2019). Analysis helps to finding functional and non-functional requirements of the project.

There are many different methods to gather information, I must use some of them like **observation**, **interview** (face-to-face and online), **focus groups** to collect information about this project called Digital Voting System.

**Some key points (reason) of using analysis:**

1. It helps to determine the exact what the system is and the requirements that perform it.
2. It helps to determine the issues (problems) in the system and provide a better solution for the problems.
3. It helps to make project success.

There are several analysis methods like **CATWOE, SWOT, PEST** etc. To find the specification requirement for the project. I must use **SWOT** analysis in this project to find the exact requirement which perform in the digital voting system project.

**SWOT** analysis stands for Strength, Weakness, Opportunities and Threats. **SWOT** analysis looks internal and external factors that can affect to the project success.

**Internal factors**

1. **Strength**: -characteristics of the website that give the advantages and help reach objectives.

**Some strength of this project:**

1. Responsive landing page/design with full mobile support
2. Quick and easy checkout process
3. Updating news and blogs
4. Contact and newsletter form to stay connected with the system
5. Intuitive navigation and search bar
6. Effective calls to action buttons
7. **Weakness**: -website characteristics that present a disadvantage or prevent from reaching objectives.

**Some weakness of this project:**

1. Less high level of security
2. Content that is not user-centric

**External factors**

1. **Opportunities**: - elements that the website could use to its advantage to help reach its objectives.

**Some opportunities of this project:**

1. New technologies to improve user experience
2. New design trends to better take messages
3. Positive changes in social factors
4. **Threats**: - elements outside of your control that could cause trouble and prevent the website from reaching its business objectives.

**Some threats of this project:**

1. New laws or regulation about voting system may be fraudulent
2. Users/voters needs changing
3. Might be copying ideas and features by competit­­­ors

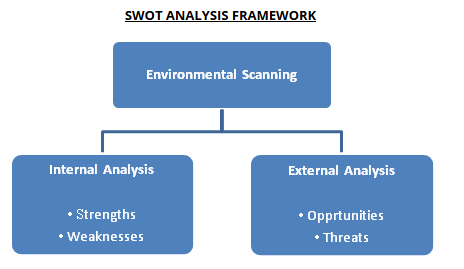


Figure 1 SWOT analysis framework

**Advantages of SWOT analysis:**

1. It discovers the strength of the system.
2. It discovers the weakness of the system.
3. It discovers any opportunities that could help to improve with the system.
4. It discovers any threats to the system.

# **Feasibility Study**

A feasibility study is an analysis that is used to determine the possibility of an idea, such as confirming a project is legally and technically feasible as well as economically correct (simplilearn.com, 2019).

This project called Digital Voting System meet all the requirements of the user, features and function also feasible and efficient. This project is better feasible and meet all the feasibilities which is given in the types of feasibility study.

**Advantages**

1. Improves the team’s concentration
2. Detects new opportunities
3. Helps decision making on the project
4. Identifies an effective reason to undertake the project

**Types of feasibility study:**

1. **Economic feasibility**: - it describes the financial resources of the project also called the cost/benefit analysis. It helps the decision making determining the positive economic benefits to the company. this project also be able and have financial resources to make this project.
2. **Technical feasibility**: - this assessment focusses on the technical resources to the organization through the processes and procedures beneficial to make project success and that is well used in this project.
3. **Schedule feasibility**: - this assessment is the most important for the project success. A project will fail if not completed on time. In this feasibility company estimates how much time the project will take to complete. Proper schedule has maintained into this project.
4. **Legal/Ethical feasibility**: -this project is fully legally to perform this assessment is the legal suggestions about the project. It is ensuring that all the requirements for the project will meet all legal and ethical.
5. **Operational feasibility: -**this assessment analyzes how a project plan satisfies the requirements and ensuring that the requirements needs can meet the project.

# **Requirements analysis**

**Functional requirements**

The functional requirements are describing the behavior of the system as it relates to the system functionality and describe what a software system should do. It includes the description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.

**Examples (features) of my project:**

1. User Registration
2. User Login
3. Calculation of total vote
4. Comments
5. Forum
6. Blogs
7. Authorization level for user
8. Cast vote
9. Party information
10. Manage candidate and parties

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **F.ID** | **Function** | **Data** | **Rational** | **Dependency** | **Remarks** |
| F01 | User Registration | personal details | User details | F01 | Create account for login into the system |
| F02 | User Login | Enter into system with Email and Password | Security | F01 | Access user dashboard and form that voters can cast vote. |
| F03 | Calculation of total vote | Total votes | Voting result | F03 | Calculate total votes for candidates that they deserve |
| F04 | Comments | User feedback | User view | F04 | User opinion about system/feedbacks |
| F05 | Forum | personal details | User details | F01 | Voters cast vote form |
| F06 | Blogs | News | News about system | F06 | Information about the system |
| F07 | Authorization level for user | For Admin (no data) | Security | F02 | Validation for unauthorized users |
| F08 | Cast vote | Voters information | Voters details | F05 | Cast vote to the specific candidate party sign |
| F09 | Party information | No data | Visiting page | F06 | Information about party |
| F10 | Manage candidate and parties | Candidates information | Update, delete, insert new candidate | F10 | Reports about candidates |

*Figure 2 functional requirements*

**Non-functional requirements**

Non-functional requirements are energetic to the success of software systems. it is describing how well the certain types of functions will work in the software systems (requirementsquest.com, 2019). non-functional requirements are the systems quality’s, attributes and the characteristics that function well in the software system.

Without this and not addressed well it will occur unwanted results like users, clients, and developers might be unsatisfied.

**Examples:**

1. Security
2. Reusability
3. Maintainability
4. Visibility
5. Reliability
6. Availability
7. Environmental
8. Data integrity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NID** | **Function** | **Data** | **Rational** | **Dependency** | **Remarks** |
| N01 | Security | N/A | To protect the significant data to make privacy and defend from unauthorized users. | N08 | Security protection should be taken to protect from hackers. |
| N02 | Reusability | N/A | For future resolution and use. | N03, N08 | System code can be reused in the future time and take as reference. |
| N03 | Maintainability | N/A | To maintain the system functions with adding some extra features. | N05, N07 | For upgrade the system with new features. |
| N04 | Visibility | N/A | The interface of the system is very user friendly and simple understandable. | N05, N07 | System interface should be interactable. |
| N05 | Reliability | N/A | The features and functions are more reliable to make easy to interact with the system. | N07 | System activities are reliable. |
| N06 | Availability | N/A | Probability of the system is not failed or undergoing a repair action when it needs to be used at availability. | N01 | System bad and good activities to make system more reputation or system failure. |
| N07 | Environmental | N/A | Interface must be a user friendly and meets all requirements. | N04, N05 | System should be more user friendly and easy to interact. |
| N08 | Data integrity | N/A | Data can be modified only by an authorized user and accuracy and consistency of stored data. | N01S | Data protection from unauthorized user. |

*Figure 3 non-functional requirements*

**Moscow priotization**

In the system, everyone wants a system always everything’s to be done right away but practically that is not possible. Therefore, reduce this problem and make proper priotization using this Moscow priotization technique. The Moscow method is about setting requirements by order of priority. The most important requirements need to be met first for a greater chance of success (toolshero.com, 2019).

Following are the Moscow priotization technique groups/categorize:

1. **M-Must have**: - It is describing the requirements that must be satisfied in the final solution. It is denoting the most vital things that can’t live without.
2. **S-Should have: -** A high priority features that is not critical to launch or the things that you consider as important, but not vital.
3. **C-Could have: -** A requirement that is desirable but not necessary.
4. **W-Won’t have: -** A requirement that will not be implemented in a current release but may be included in a future stage of development. Such requirements usually do not affect the project the project success. A requirement that provide little to no value you can give up on.

**Prioritization of functional requirements:**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **PRIORITY** |
| FR01 | User Registration | Must have |
| FR02 | User Login | Must have |
| FR03 | Calculation of total vote | Must have |
| FR04 | Comments | Should have |
| FR05 | Forum | Should have |
| FR06 | Blogs | Should have |
| FR07 | Authorization level for user | Must have |
| FR08 | Cast vote | Must have |
| FR09 | Party information | Should have |
| FR10 | Manage candidate and parties | Must have |

*Figure 4 priorities of functional requirements*

**Priority of non-functional requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **PRIORITY** |
| NFR01 | Security | Must have |
| NFR02 | Reusability | Must have |
| NFR03 | Maintainability | Must have |
| NFR04 | Visibility | Must have |
| NFR05 | Reliability | Should have |
| NFR06 | Availability | Should have |
| NFR07 | Environmental | Must have |
| NFR08 | Data integrity | Must have |

*Figure 5 priority of non-functional requirements*

**SRS (software requirement specification)**

A software requirements specification (SRS) is a document that describes what the software will do and how it will be likely to perform. A software requirements specification is the basis for your entire project. It arranges the background that every team involved in development will follow.

**Software requirements for this project: -**

**Programming language: -** PHP core with MVC Pattern

**For class diagram and use case diagram: -** StarUML

**Operating System: -** Windows 10 Enterprises with 4 GB Ram

**For documentation: -** MS Word 16

# **Use case diagram**

A use case diagram is a graphical representation of the interaction between the elements of a system (whatis.techtarget.com, 2019). It is a method used in system to investigation detect, explain and organize system requirement. Use case showing the relationship between the user and the different use cases in which the user is involved in the system.

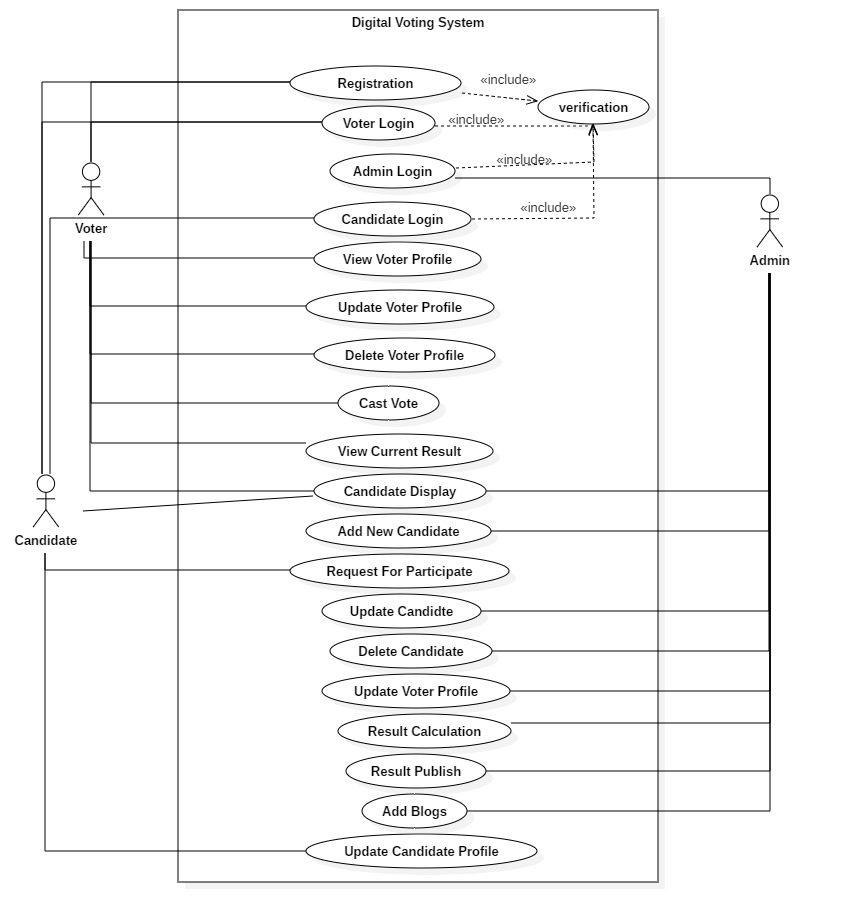


Figure 6 use case diagram of Digital Voting System

**Scenario of Digital Voting System**

Some of the countries apply the election system online to make secure and more efficient but Often must of the countries have traditional rules for election that makes people feel more stress to stand on long queue to give a vote.

I must have a plan to build an Online Voting System (Online Election System) as a web application to make people easier to cast vote and provide a high level of security to protect the voting System. In this system, voters can register/create their account and then login to the system in which voters can cast vote, update their profile.

Some activities done by the main character in this project:

**Voters: -** register and login, cast vote, (update, delete, view their profile), view candidate list.

**Candidate: -** register and login, view candidate list, update their profile, request for participate.

**Admin: -** admin login, view candidate list, (update, delete and add new candidate), result calculation, result publish and add blogs.

# **Natural Language Analysis**

Natural Language Analysis is the process of identifying verb, adjective and candidate class. I must search some nouns related to the possible classes and verbs related to the possible functionality for class diagram from the given scenario.

1. **Nouns and verbs that are found in the given scenario:**

|  |  |
| --- | --- |
| **Nouns** | **Verbs (Method)** |
| Countries, election system, countries, rules, traditional, election, people, vote, plan, online voting system, web application, people, cast, high level, security, voting system, system, voters, account, system, voters, vote, participate, candidate, profile, admin, candidate, new, result, blogs. | Apply, make secure, give a vote, cast vote, build voting system, easier to cast vote, provide security, protect system, register account, create account, cast vote, update candidate profile, delete candidate, register and login, cast vote, update voter profile, delete voter profile, view voter profile, view candidate list, request for participate, result calculation, result publish, add blogs. |

*Figure 7 NLA (natural language analysis)*

1. **Candidate class and verbs**

|  |  |
| --- | --- |
| **SN** | **Selected Nouns** |
| 1 | Voter |
| 2 | Candidate |
| 3 | Admin |
| 4 | Result |

*Figure 8 candidate class and verbs*

1. **Potential verbs**

|  |  |
| --- | --- |
| **SN** | **Selected Verbs (Methods)** |
| 1 | Cast vote |
| 2 | Update candidate profile |
| 3 | Delete candidate |
| 4 | View candidate list |
| 5 | View voter profile |
| 6 | Update voter profile |
| 7 | Delete voter profile |
| 8 | Result calculation |
| 9 | Request for participate |
| 10 | Result publish |

*Figure 9 potential verbs*

# **Class diagram**

A class diagram is a type of diagram and part of a unified modeling language (UML) that defines and provides the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes.

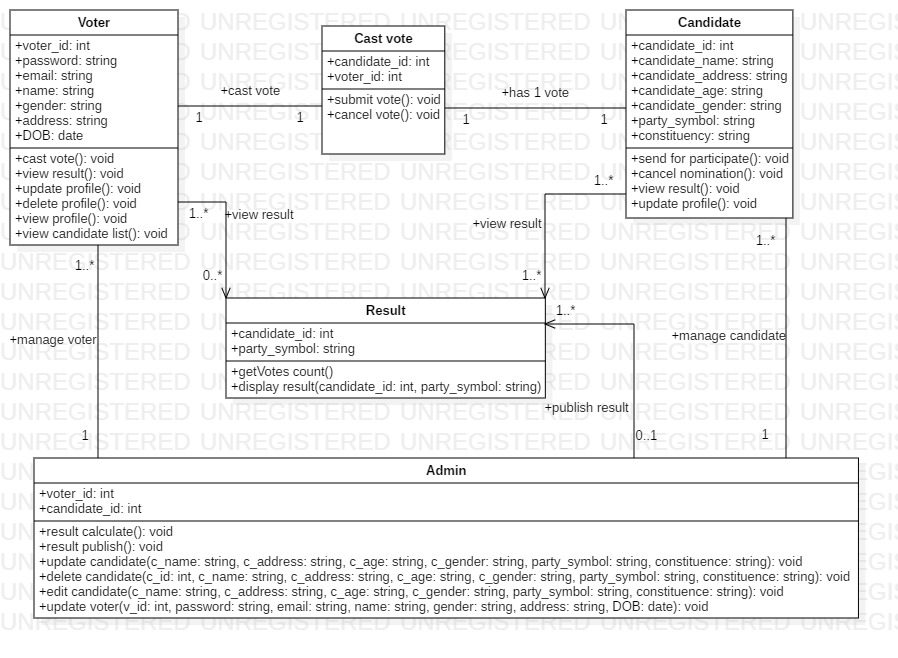


Figure 10 class diagram of digital voting system