# Software Requirements Specification

# for

# Game Factory for educating Illiterate people

Version 1.0 approved

Prepared by Kanv Kumar, Syam Kumar Patnala, T Bharat, Sai Teja

IIIT Sricity

25 th September 2015

Table of Contents

[1.Introduction 1](#__RefHeading___Toc441230972)

[1.1Purpose: 1](#__RefHeading___Toc441230973)

[1.2Document Conventions 1](#__RefHeading___Toc441230974)

[1.3Intended Audience and Reading Suggestions 1](#__RefHeading___Toc441230975)

[1.4Product Scope 1](#__RefHeading___Toc441230976)

[1.5References 1](#__RefHeading___Toc441230977)

[2.Overall Description 1](#__RefHeading___Toc441230978)

[2.1Product Perspective 1](#__RefHeading___Toc441230979)

[2.2Product Functions 2](#__RefHeading___Toc441230980)

[2.3. User Classes and Characteristics 2](#__RefHeading__919_173753190)

[2.4 Operating Environment 2](#__RefHeading___Toc441230982)

[2.5 Design and Implementation Constraints 2](#__RefHeading___Toc441230983)

[2.6User Documentation 2](#__RefHeading___Toc441230984)

[2.7Assumptions and Dependencies 2](#__RefHeading___Toc441230985)

[3External Interface Requirements 2](#__RefHeading___Toc441230986)

[3.4 User Interfaces 3](#__RefHeading___Toc441230987)

[3.5 Hardware Interfaces 3](#__RefHeading___Toc441230988)

[3.6 Software Interfaces 3](#__RefHeading___Toc441230989)

[3.7Communications Interfaces 3](#__RefHeading___Toc441230990)

[4System Features 3](#__RefHeading___Toc441230991)

[5Other Nonfunctional Requirements 3](#__RefHeading___Toc441230994)

[5.4Performance Requirements 3](#__RefHeading___Toc441230995)

[5.5 Safety Requirements 3](#__RefHeading___Toc441230996)

[Security Requirements: 4](#__RefHeading__921_173753190)

[5.6Software Quality Attributes 4](#__RefHeading___Toc441230998)

[5.7Business Rules 4](#__RefHeading___Toc441230999)

[6Other Requirements 4](#__RefHeading___Toc441231000)

Revision History

To be updated....

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose:

The Existing System of Election is running manually. The Voter has to Visit to Booths to Vote a Candidate so there is wastage of Time. The Voter has to manually register into the Voter List. Also Vote counting has to be done manually. So online voting system can be used to minimize these problems. The uses of it are

1. It Maintains all The Information of all the Candidates and Votes.
2. It checks Voter have Voted or Not.
3. It Increases the Voting Percentage

## Document Conventions

Italic style : We use this for highlighting the technical points in the document.

Bold style : Heading in the document.

## Intended Audience and Reading Suggestions

This application doesn't need any prerequisites to learn. In the further documentation I have explained in detail how to use this application and also the complete description about the system.

## Product Scope

The Scope of the System is as Follows:

1. Voter can Vote from anywhere for his/her Constituency.
2. Vote count will make easy and fast.
3. No any Vote will be rejected.

## References

Source code of some of the basic system which are already present but not affective.

Use of “stackoverflow” website for getting some of the doubts cleared.

# Overall Description

## Product Perspective

This system will be made up of two parts, one running visible directly to the administrator on the server machine and the other visible to the end users, in this case the voters, through web pages. The two users of the system, namely the voters and the election admin interact with the system in diﬀerent ways The voters cast their votes using the web interface provided. These votes are accepted by the system on the server.

## Product Functions

The system can be used to create/update/delete the election details ( posts, candidates, electoral rolls etc ). The admin should be able to specify the diﬀerent attributes it wants for posts/candidates of a particular election instance and voters. The system should also be able to run seperate election instances at the same time.

## 2.3. User Classes and Characteristics

The users can be divided into two main classes: –

The Admin :

It’s primary objective is to conduct fair elections. The admin has to be a neutral party and should not have any gain/loss from the election results. The admin decides the classes of voters eligible to vote for a certain post. They should have adequate experience of using a computer to be able to conﬁgure the election properly.

The Voters : The voters should have a basic knowledge of how to use a web browser and navigate through web pages. The voters should be aware that they have to keep their user-id and password conﬁdential.

## Operating Environment

The server should have Java installed on the machine, along with Java’s cryptographic packages. The election server runs on a http server . The browsers through which the voters access the server should have minimal support for cookies and encrypted transactions.

## Design and Implementation Constraints

We are using Java, JavaScript or php and MySQL

## User Documentation

To be updated.....

## Assumptions and Dependencies

--PC (Personal Computer) or workstation with GUI.

– A web browser with support for cookies.

– Working Internet connection

– A web server with GUI, Java and an http server installed

# External Interface Requirements

## User Interfaces

Use case 0 : Welcome screen for the Administrator

Use case 1 : Creating the Voters Database

Use case 2 : Modify the voters databases

Use case 3 : Delete the Voters’ database

Use case 4 : Add the Candidates

Use case 5: Modify the candidates

Use case 6 : The Voting on the Voter’s end

Use case 7: Counting the votes

Use case 8: Declaring the results

## Hardware Interfaces

There are no hardware interfaces to this software system. The only interfaces are through a computer system.

## Software Interfaces

The poll server runs on http server. It uses a relational database to keep track of the polls. In order to run the setup software, the environment needs to have a JVM running on it.

## Communications Interfaces

The system should use standard protocols for secure transactions between the Voter and the system through the internet.

# System Features

To be updated…….

# Other Nonfunctional Requirements

## Performance Requirements

The software is expected to have reasonably short response time. It should be able to log-in and feed the voter with new pages on request with a response time of the order of a few seconds.

## Safety Requirements

In order to prevent data loss in case of system failure, the result of votes that were polled till then have to be saved in the database. The system should be capable of gracefully recovering from earlier crashes and continuing the voting process.

## Security Requirements:

The system should provide basic security features like password authentication and encrypted transactions. All the passwords generated and communicated to the users should be stored in the server only in an encrypted form. Aadhar cards should be linked and OTP also should be used.

## Software Quality Attributes

The code which will be building is scalable in terms of code modifications, code enhancements.

Also the code build can be extended for further use in different modules in the project.

Testing the software for bugs and fixing them. Also adapting Agile Methodology like Scrum, we can build better software faster.

.Moreover, the final product would be robust in terms of quality and performance.

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

If the game is having some glitches, then the user can contact the developers for their assistance and thus bugs might be solved.

Feedback from the users will be appreciated.

# Other Requirements

The database which we create should be structured so that the query time should be less.

The database should be normalized before running the queries.

Appendix A: Glossary

To be updated....

Appendix B: Analysis Models

To be updated....

Appendix C: To Be Determined List

To be updated....