**3.2.2 Purpose**

The purpose of this document is to provide Software Requirement Specification for a Secure approach for Web Based Internet Voting System using Multiple Encryption. The software requirement specification document enlists all necessary requirements for project development.

**3.2.3 Scope**

The software product produced is an application by name “BBMP Online Voting Portal with Multiple Encryption (www.vtuproject2015.com/vB)".

This project suggests a practical application of the existing cryptographic schemes and digital signature that ensures integrity of the vote cast by voter and authentication of voter at the two levels. Design of secure e-voting system over a network is indeed a very difficult task as all the requirements of the voting system have to be met. Failure to ensure even one of the specifications can lead to chinks and glitches that can be exploited by a middleman to forge or manipulate the intricate details. Subsequently, the result of the election is computed from the sum of the votes which is jointly decrypted by the authorities. A voting scheme must ensure that the voter can keep his vote private.

**3.2.4 Definitions, Acronyms, and Abbreviations**

**Various divisions in the project**

* **Voting Ballot**

This uses the user interface for logic by the user which is designed using twitter bootstrap and foundation frame work. Then the deployment descriptor does the mapping of the requests to the existing servlets in the backend. Then comes the layer which performs registration, login , logout, Casting a vote, entering the OTP Database layer to store the user interface information for authentication purpose like OTP.

* **Voting Server**

The user interface only for the admin which is designed using the twitter bootstrap an foundation framework Deployment descriptor does the mapping of the requests to the existing servlets in the backend. Then comes the layer which has operations such as—New election process, Participate, results of the election, Authentication, Other admin operations and also security Database layer to store the data regarding the election process

**3.3 General Description**

**3.3.1 Product Perspective**

* Implementation of Voting Ballot where the citizens of the assembly will be using this portal for various functionalities
* Implementation of Voting Server where the election commission will be using this portal for various functionalities
* J2EE container like Jboss, or glassfish or the Jetty, or the Websphere application server (WAS) to host the interface layer developed.
* RSA algorithm to generate the keys used for encryption and decryption of the vote.
* DES encryption technique.
* RSA digital signature algorithm for digitally signing the vote that will be sent from voting ballot to voting server.
* The end user’s machine accessing the application from the remote location need not be installed with the RHEL. It can be any operating system

**3.3.2 User Functions**

* The client of this project will be accessing the interface layer of the Voting Ballot from remote location. They will not be given the direct access (neither to any of the services on Voting Server).
* The admin of this project will be accessing the interface layer of the Voting Server from remote location. He will not be given the direct access to voting ballot.
* In the current application in consideration, the user will be given with a simple User Interface developed using HTML, CSS, and JavaScript, and Foundation framework.
* The User Interface provided to the client will have multiple sub modules like voting, participants, analytics, etc

**3.3.3 Product Characteristics**

* A small amount of configuration work to be done in the RHEL box to set up the Voting ballot and Voting server
* JBoss AS 7.1.1 version Application server to host the applications.

**3.3.4 Assumptions and Dependencies**

* JDK has to be installed in all the machines where the Voting Ballot and the Voting Serveris deployed and also where the map reduce program has been executed.
* The application servers like either the JBOSS or the Apache Tomcat will have to be supported by the host machines
* There shall not be any firewall or other engines that prevents the remote requests from the interface portal.
* The RHEL box on which the application cluster will be configured will be up and running all the time to avoid the outages.
* There shouldn’t be any permission related issues on the Hadoop cluster. The host operating system should take of permitting all the requests to the Hadoop cluster from the interface layer.

**3.4 Requirements**

**3.4.1 Functional Requirements**

* User should be able to securely login the application through one time password verification technique.
* User should be able to upload the ID proof document to get the access to vote.
* User should be able to view all the elections that are scheduled.
* User should be able to vote securely for a scheduled election which he belongs to the same assembly.
* User should be authenticated again through one time password verification technique during the time of voting as well.
* User and Admin should be able to view the results soon after the elections are terminated or ended.
* Admin should be able to view or verify the ID proof documents uploaded by the users and take a call on approving or rejecting them.
* Admin should be able to create / delete / update any elections and add participants to it.

**3.4.2 Non Functional Requirements**

* Should be easier to access it from the various browsers available.
* Response time of the applications should reflect the real time observations.
* The algorithm should never fail in any of the test cases.
* Each user’s activity should be separated from the other user’s activities

**3.4.3 Software Requirements**

* Operating System: Windows XP or higher
* JDK 1.6
* Any latest Applications server like JBoss, Glassfish
* Eclipse

**3.4.4 Hardware Requirements**

* Processor: Intel Pentium 4 or higher
* RAM: Min 512MB
* Hard Disk: 40GB