CleanVote

System and Software Architecture Description (SSAD-LCO)

Team 2

2011120240 박종훈

2012190720 황서연

2012190706 허능호

2012210042 박채흠

2013171058 김재훈

2013220053 오정민

2014120045 이건선

Table of Contents

Version Control

Date	Author	Changes	Version
09/28/2016	박채흠	Initial version	0.1
10/25/2016	박채흠 황서연 김재훈	Added UML Diagrams	0.2
11/09/2016	오정민 박채흠	Minor Changes	0.3
12/14/2016	이건선	Final Check up	1.0

List of Figures

1 Introduction

1.1 Purpose

The System and Software Architecture document analyzes the Block-chain based voting system requirements and presents an overview of the architecture and design of the proposed system.
Here, top-level definition of at least one feasible architecture for the station data project will be presented along with the identification of logical and physical elements and relationships.

1.2 Standards and Conventions

1.2.1 Notation Used

UML Notation will be strictly followed throughout this document.

1.3 References

- System and Software Requirements Definition Document 1.0
- Group Meetings
- Group Repository(Github)

2 Architectural Analysis

This section will provide a high level solution of the Block-chain voting Problem. This will be done through identification of the components and behaviors of the proposed system. This architecture analysis can then be used for implementing the system.

2.1 Component Model

The proposed system consists of the following components:

2.1.1 COM-01: 'Web App'

The web interface component is the user interface of the 'CleanVote' (Block chain based online voting system) on the World Wide Web. It consists of a set of web pages that provide the user of the system an input screen for entering candidate, signing key and an output screen that displays the poll results. There are a few more pages that are part of the web interface like the demo try-out, and contact pages in order to create a poll for clients.

Component Identifier	COM-01
Defining Quality	The user interface that is available on the world wide web for block-chain online voting.
Name	Web Interface
Attributes	a) Web URL. b) Candidate information c) Signing key entry area.
Behaviors	a) Display Certain poll information page. b) Display Candidate information c) Get signing key and candidate selection and send to BigchainDB and MySQL Public Information. d) Display search results on the web page.
Relationships	a) BigchainDB Node (COM-02). b) MySQL Public Information (COM-03)
Roles	a) 'Get Input', 'User'. b) 'Display Results', 'User'. c) 'Send Voting Info', 'BigchainDB Node'. d) 'Send Voting Info', 'MySQL Public Information'. e) 'Retrieve Results', 'MySQL Public Information'.

State Groups	a) BigchainDB Node {Sending}
	b) MySQL Public Information {Sending,Retrieving, Displaying}

Constraints	TBD.

Relates To	

2.1.2 COM-02: ‘BigchainDB Node’

The BigchainDB Node component in the system is an Open API for basic block chain functions such as creating keys and transferring assets. This is the central processing unit of the system that receives input, which is the selected candidate and signing(private)key of a user from the web interface, and then conducts transaction based on that information.

Component Identifier	COM-02
Defining Quality	The central processing unit of the station data search system that receives inputs from the web client interface and conducts transaction of the asset from the user to the candidate and then sends the information to the MySQL Public Information.
Name	BigchainDB Node
Attributes	
Behaviors	a) Receive transaction request from web interface. b) Check if the ballot(asset) is valid c) Transfer ballot(asset). d) Send results to MySQL Public Information
Relationships	a) Web Interface (COM-01). b) MySQL Public Information (COM-03).
Roles	a) 'Receive Transaction Request', 'Web Interface'. b) 'Asset Transaction', 'Systems Components'. c) ‘Check Validity’, ‘Systems Components’. d) 'Send Results', 'MySQL Public Information'.
State Groups	a) Web Server {Receiving, Processing}. b) MYSQL Public Information {Sending}
Constraints	TBD.
Relates To	

2.1.3 COM-03: ‘MySQL Public Information’

MySQL Public Information is the component that is responsible for matching the inserted signing key with containing public keys, thus validating the user. Also It stores the ballot that came from BigchainDB Node. When the time of the poll is over, the Web interface will check the ballot table to count the results.

Component Identifier	COM-03
Defining Quality	This component is responsible for matching the inserted signing key received from the web interface with the public key that it retains and validating the user. It also stores ballot transaction information from the BigchainDB Node.

Name	MySQL Public Information.
Attributes	
Behaviors	a) Receive signing key from the web interface. b) Search for matching public key in database. c) Store transactions sent from BigchainDB Node.
Relationships	a) Web Interface (COM-01). b) BigchainDB Node (COM-02).
Roles	a) 'Receive User Input', 'Web Interface'. b) 'Match key', 'MySQL Public Information'. c) 'Store information in tables', 'BigchainDB Node'.
State Groups	a) Web Interface {Receiving}. b) BigchainDB Node {Receiving}
Constraints	TBD.
Relates To	

The Component Model is shown below: (specific relationship will be shown in the last flowing chart diagram)

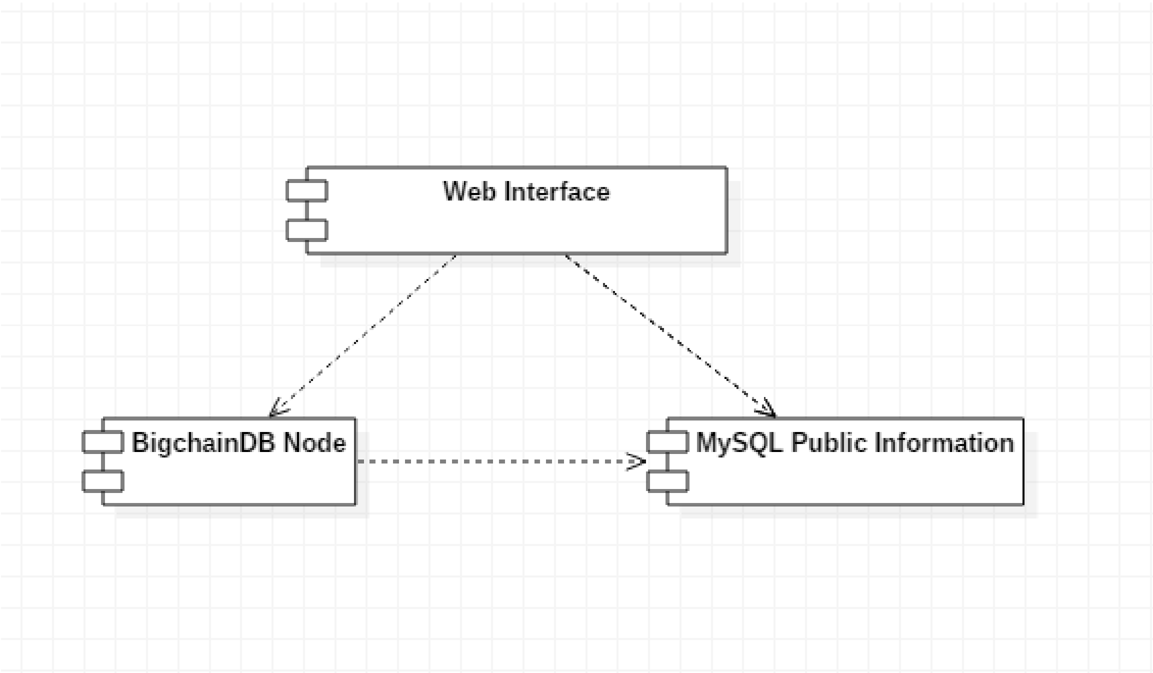


Figure 1 Component Model

2.2 Behavior Model

The following are the system's behaviors:

2.2.1 BH-01: 'Cast Vote'

This behavior is associated with the web interface and is triggered when the user clicks on a certain candidate among filling in its signing key into the text area and then clicks submit.

Behavior Identifier	BH-01
Trigger	User clicks on the submit button.
Preconditions	The user fills out signing key, selects a candidate.
Post conditions	A message acknowledging the vote casting action
Inputs	Data of signing key, selected candidate info.
Outputs	A pop up message
Exceptions	Invalid Voter or Invalid ballot.
Sequence Diagram	Refer to Figure 6.
Relates to	
Type	

Capability	System should be able to verify voter and store voting information.
System Sub-Responsibility	The system should allow the user to input signing key and to select among candidates.
System Behavior	System checks for error.

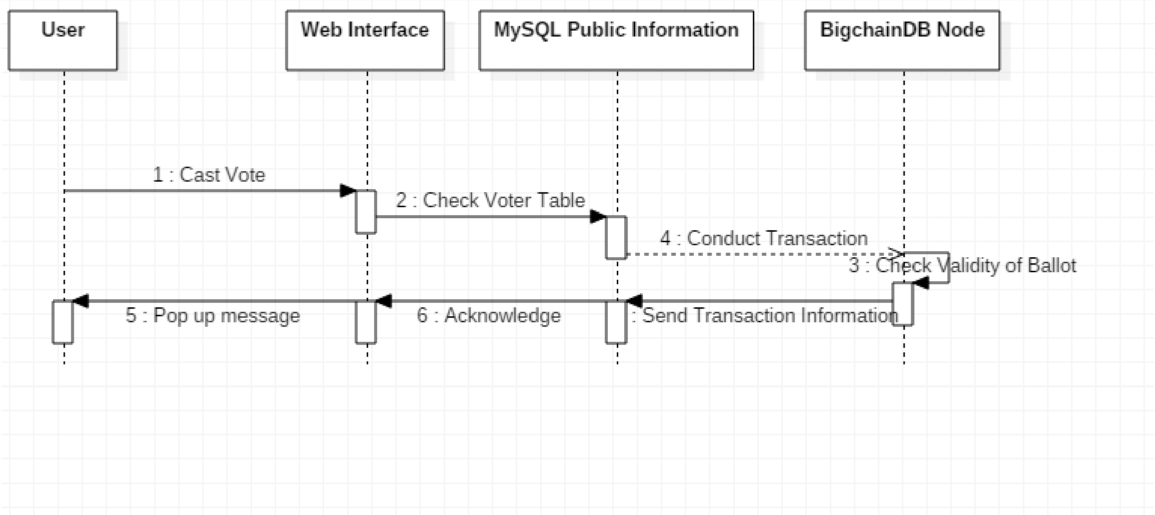


Figure 2 BH-01: Cast Vote

2.2.2 BH-02: ‘Display results’

This behavior is also associated with the web interface. Once poll has reached to its closing time the Web Interface will retrieve data from MySQL Public Information and count, display the result for users to see.

Behavior Identifier	BH-02
Trigger	Poll time is over
Preconditions	Poll should have been properly created with necessary information.
Post conditions	The results are saved and displayed for users.
Inputs	Ballot table
Outputs	A screen displaying results including list of voters’ public keys.
Exceptions	The results should not be retrieved by users before closing time.
Sequence Diagram	Refer to Figure 7
Relates to	
Type	

Capability	The results should show proper information on ballot tables.
System Sub-Responsibility	The web interface allows the user to see the results and the public keys and selected candidates from each.
System Behavior	A screen is generated containing the results.

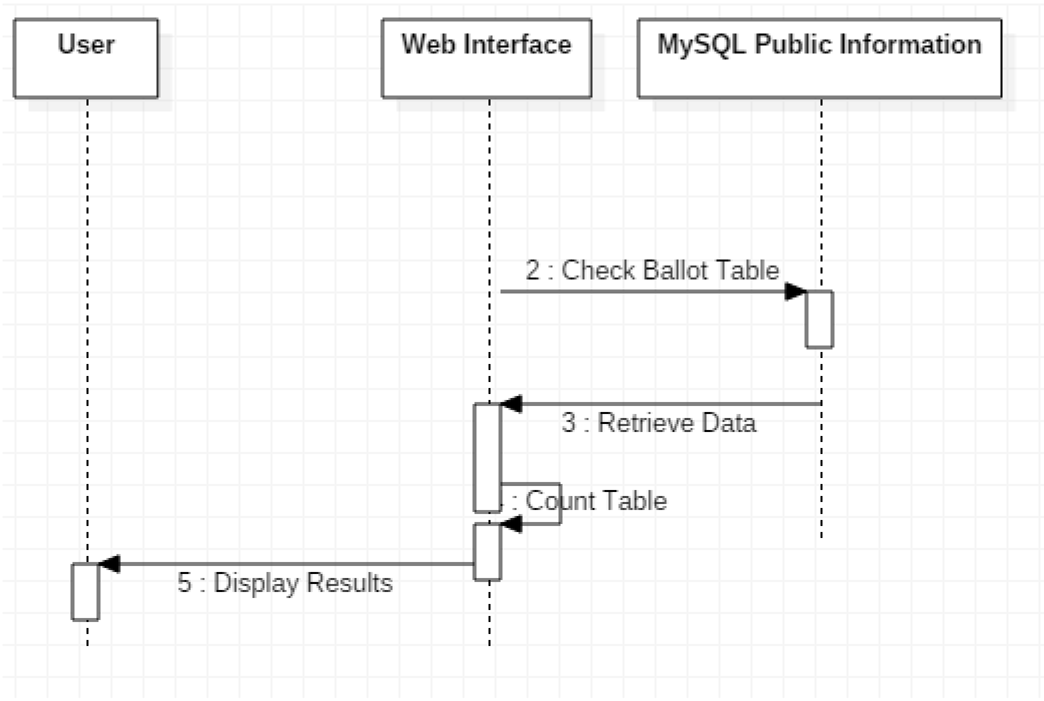


Figure 3 BH-02: Display Results

2.2.3 BH-03: ‘Create Poll’

This behavior is associated with the both BigchainDB Node and MySQL Public Information. Once the client has sent all the necessary information for a poll, the BigchainDB Node will generate keys to distribute to the selected users. Private keys will be sent to users’ emails which is from MySQL Public Information’s user table. The private keys will be discarded immediately. Yet the public keys will be stored in MYSQL Public Information.

Behavior Identifier	BH-03
Trigger	Client requests poll creation.
Preconditions	The client should submit an excel file containing user name, student ID, phone number, email.
Post conditions	The selected users will receive a text message or an email containing the URL and an assigned signing key.
Inputs	Number of users required in the poll.
Outputs	Email or text message to the user.
Exceptions	
Sequence Diagram	Refer to Figure 8
Relates to	
Type	

Capability	The system maintains user information security by discarding the generated private key information right after sending it to the usrs
System Sub-Responsibility	MySQL Public Information should delete public keys.
System Behavior	There is no information of matching of a certain user and a signing key. Only the public key remains.

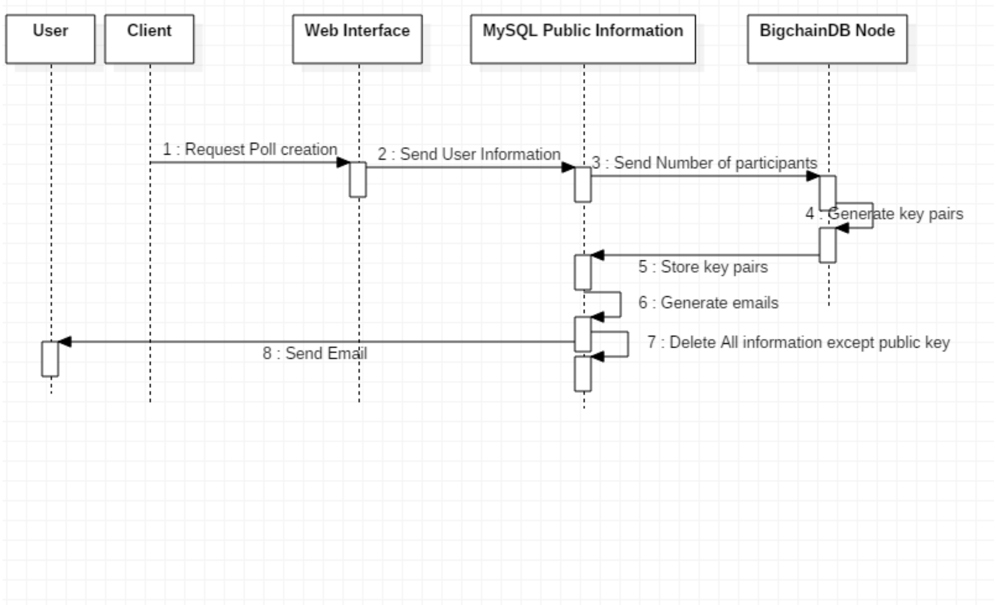


Figure 4 BH-03: Create Poll

2.3 Enterprise Model

2.3.1 Component Classifications

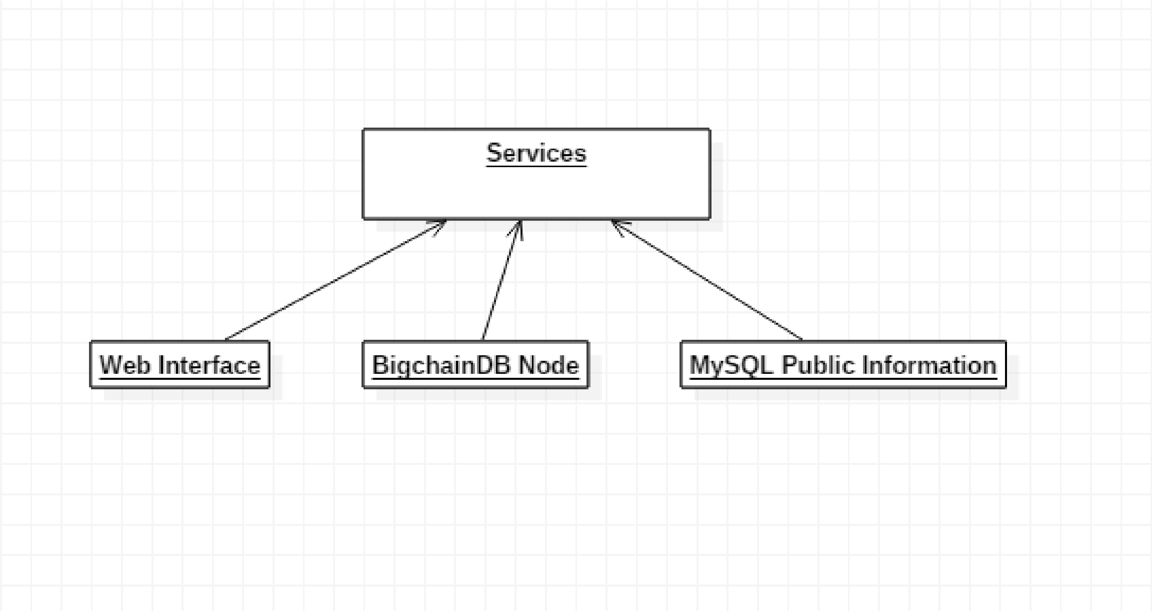


Figure 4 Component Classifications

2.3.2 Behavior Classifications

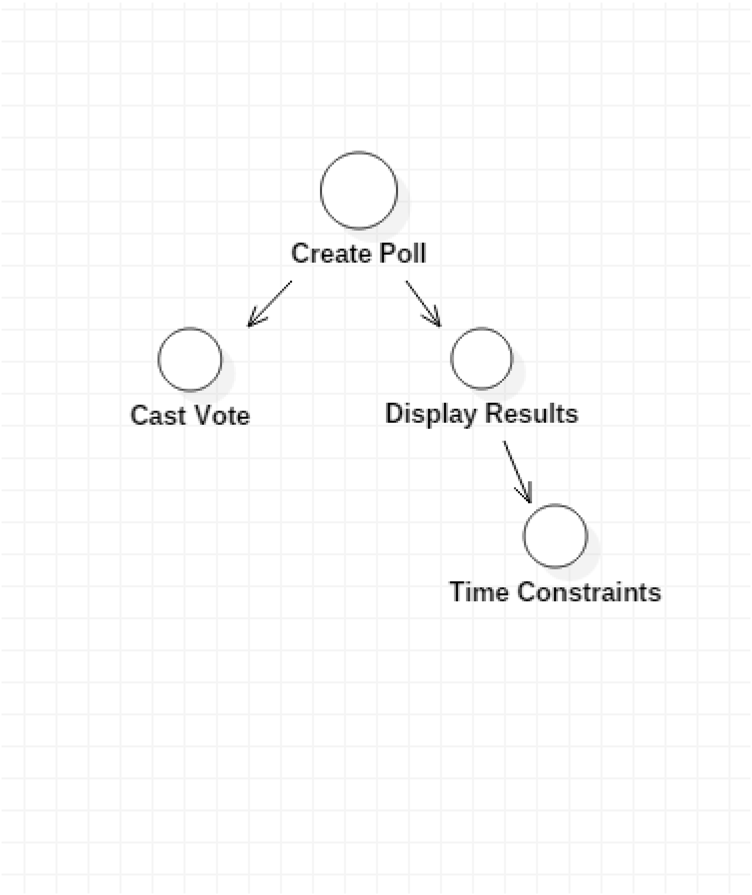


Figure 5 Behavior Classifications

3 System Design

3.1 Architectural Views

3.1.1 System Topology

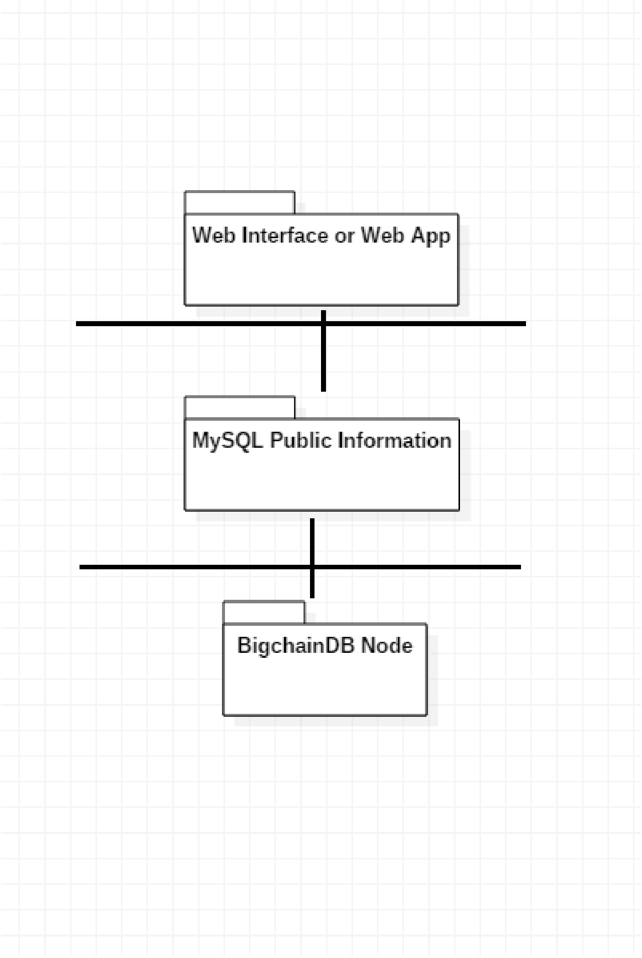


Figure 6 System Topology

3.1.2 Component Specifications

Component Identifier	DCOM-01
Defining Quality	The template for the web interface pages that are displayed on the internet for input and results.
Name	Web Page Template
Attributes	a) Header and Footer b) Background Color and Font
Behaviors	a) Display web pages.
Relationships	
Roles	a) Format Web Page
State Groups	a) Web Template {Filled, Empty}
Constraints	TBD.
Implementation	HTML,CSS3,Javascript

Component Identifier	DCOM-02
Defining Quality	Application that runs the system.
Name	Web Server
Attributes	a) Server IP Address
Behaviors	a) Run web search application
Relationships	a) BigchainDB Node – COM-02
Roles	a) Web Application Manager
State Groups	a) Web Server {Running, Idle}
Constraints	TBD.
Implementation	AWS-EC2

Component Identifier	DCOM-03
Defining Quality	Open API that creates needed functions.
Name	BigchainDB
Attributes	
Behaviors	a) provide key generation functions b) provide asset transaction functions
Relationships	a) BigchainDB Node – COM-02
Roles	a) Provide block chain functions
State Groups	a) BigchainDB Node {Running, Idle}
Constraints	TBD.
Implementation	BigchainDB

3.1.3 Framework and Protocol Specifications

- AWS-EC2 is used for the development of web server.
- CSS3, Javascript is used for the generation of dynamic web pages.
- BigchainDB is a block chain based database that provides basic block chain functions.

3.1.4 Logical Component View

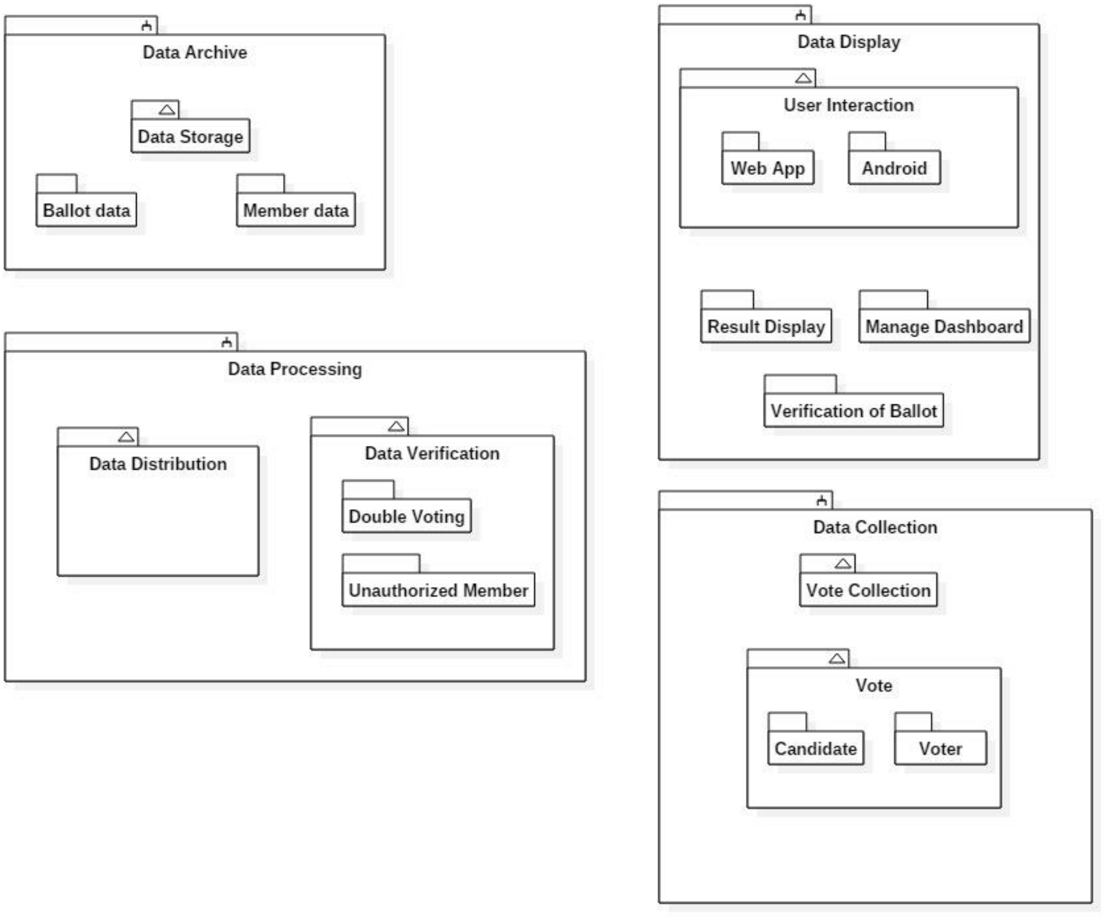


Figure 7 Package Diagram

3.2 Class Model

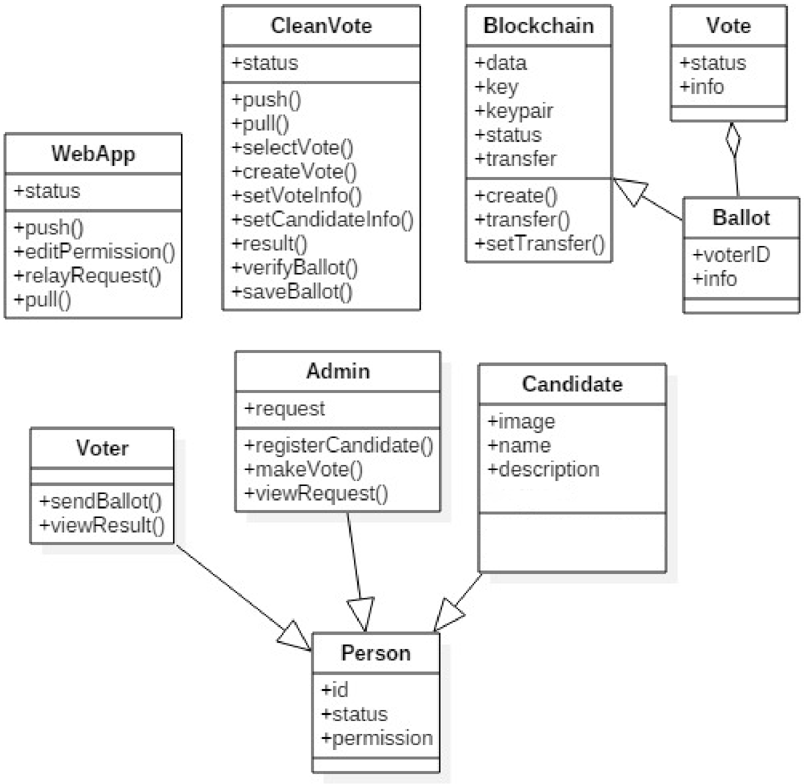


Figure 8 Class Diagram

4 Common Definition Language for System Design

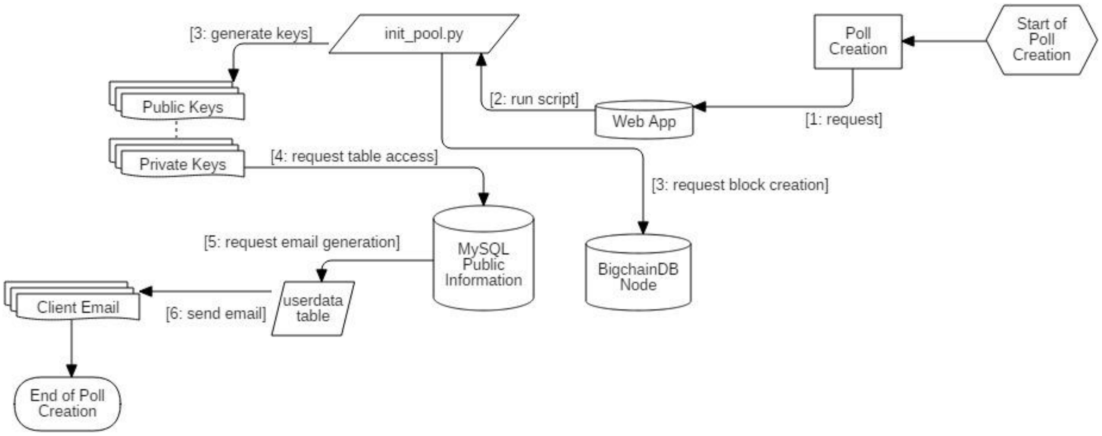
- HTML(Hyper Text Markup Language)
  HTML is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS), and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.
- CSS3(Cascading Style Sheets 3)
  CSS3 is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics.
- JavaScript
  JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented,[8] imperative, and functional programming styles.

- JDBC (Java Database Connectivity)
  Component used to connect to various database resources such that the connection is transparent to the implementation.

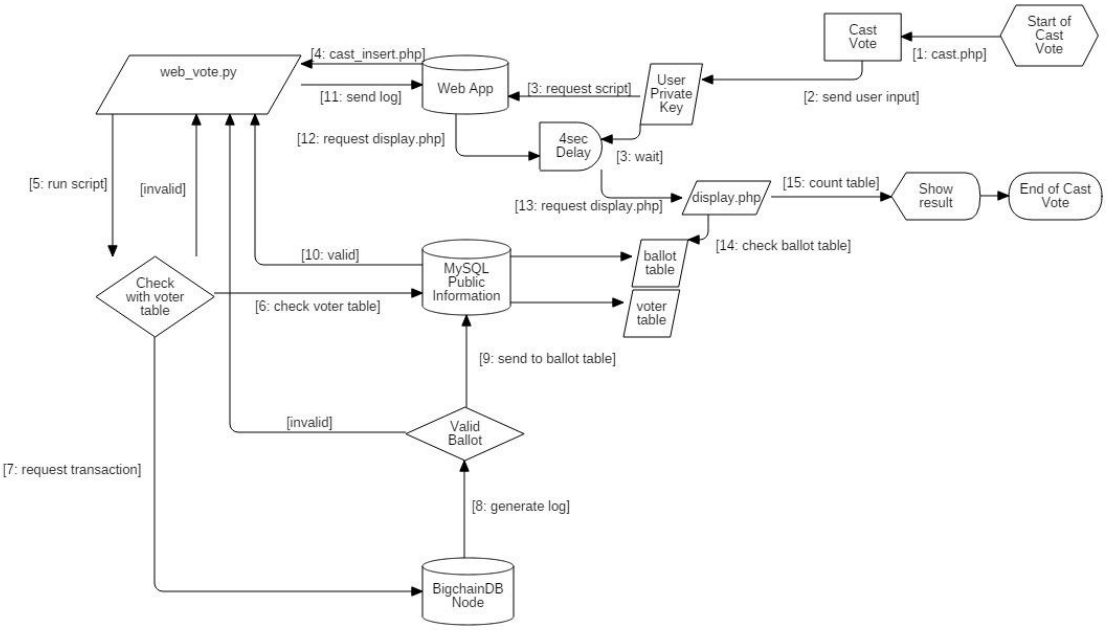
- BigchainDB
  bigchainDB is an Open source API that allows developers and enterprise to deploy blockchain proof-of-concepts, platforms and applications with a scalable blockchain database, supporting a wide range of industries and use cases.

5 Appendices

- Flow Chart of poll creation



- Flow Chart of Vote Casting


Start of Cast Vote
[1: cast.php]
Cast Vote
[2: send user input]
User Private Key
[3: request script]
Web App
[3: wait]
4sec Delay
[4: cast_insert.php]
web_vote.py
[11: send log]
[12: request display.php]
[13: request display.php]
display.php
[15: count table]
Show result
End of Cast Vote
[14: check ballot table]
ballot table
voter table
[5: run script]
[invalid]
[10: valid]
Check with voter table
[6: check voter table]
MySQL Public Information
[9: send to ballot table]
[invalid]
Valid Ballot
[7: request transaction]
[8: generate log]
BigchainDB Node