**Name:** Ikran Hassan, Dion Chever, Shemeles Abebe

## Original Design

At the beginning stages, the assumption was that the database had to cover all different types of elections including the federal, state, and local elections so the database was setup to cover all three types of election.

The initial design would have implemented the functionality using a fingerprint scanner and conceptually facial recognition software. The facial recognition feature was intended for disabled voters who had mobility or tactile impairments. On the other hand, the fingerprint table was created to store fingerprint images to account for voter who are identical twins. In theory when they registered to vote a facial image is stored in the database, and used as a comparison during time of vote. The same functionally concept was applied to fingerprint scanners. The users would logging into their account to vote, and upload a fingerprint image. By using facial recognition they could just scan their face and log into their account. The database would hold the original facial image scanned when they register to vote. Next, when they log into the system by using their unique username and password, a facial scan is needed again and the database would compare the two images and check if they are a match. We did not want the facial recognition software in the database to be fooled, so we wanted to use fingerprints because everyone's fingerprint is different. The fingerprint functionality was intended to works in the same way as facial recognition. The deceased table was created to stop people from using a deceased person SSN to try and register to vote. So once the admin user receives information that a person has passed away, the deceased person SSN, first and last name, and date of death will be moved into the deceased table. Whenever a person registers to vote the

trigger assigned on the person info table checks the SSN provided by the user with the SSNs stored in the deceased table and prevents identity theft of deceased people.

The initial design for holding voter information was meant to hold one user's information into four different tables: useraccounts, person info, resident info and voter. The user accounts included the user's first and last name, username, password, Social Security Number, date of birth, citizen number and if they were a twin. We also had a separate table for person info which included the same information with some additional information like resident id, ethnicity, gender, race, voter id, and if they were disabled. Once a person was inserted into the person info table, we were going to insert them into the resident info table with their address information.  The resident id was established in the person info table and use their voter id that we get after they are inserted into the voter table. Then insert their address including street, city, county, state, and zip code into the resident info table. The voter's table is created to give a voter their voter id and keep track if they voted. It was designed to hold the voter id, the voter's full name their citizen number and a boolean value of if they voted in the current election.

The people who became a registered voters need a place to vote in a location close to them so we created the district info and polling station, which combined tells a person where the polling station is in a state. District Info table was created to store the voters id who voted for a specific candidate with regards to the state, city, and zone. The setup behind the polling station was to store the voter id, candidate id, the state, city and zone the polling station is located.

The people who are registered to vote and going to the polling station need candidates to vote for, so we created a table for the parties and candidates, each party name includes a party id. The candidates table was designed to store the running candidate unique id, their name, party affiliation, term they are running, and what state the candidate is from.

Now that we have registered voters, a place for them to vote and candidates for we need

to store their votes in the ballot table. The ballot table was to store registered voters information, who voted a specific candidate including the county and zone information of the assigned ballot. To store past ballots, a ballot history table was created to hold any ballots that held past ballots instead of the current election. The ballot history held the history ballot id and all the same information that is in the ballot table.

## Final Implementation

We started Developing and Coding the front-end Graphical User Interface with a visual Studio .Net and Visual Basics. Only after creating the entire GUI and determining how it would look, we realized connecting the mysql database with the designed page would be harder in VB. After seeking GUIdance from Dr. Wilson, we started learning HTML and PHP to connect the database and create both html/php page. We used the latest version of Xampp, and Apache Server, to connect the GUI and the database.

The scope of the database is enabling eligible citizens to cast their vote for the general presidential election candidates for any given term through a secure application database system. The fingerprint scanner feature was not working for us so we decided to drop the entire table and any columns that reference it in any table including the person info and ballot table.

The deceased info table holds the SSN of people who have passed away with there name and a date of death. There is a trigger that checks if the SSN is in the deceased table then do not allow the user to register to vote with that SSN and checks the date of birth to make sure the user is at least 18 years or older when they are registering.

Once we started to implement people into our database we ran into issues with foreign key constraints and table hierarchy. We were not able to insert information into any of the tables because the other tables did not have any information so we had to go back and combine tables

and remove column from tables. We ended up going back removing the user_accounts table all together and moved username, password, Social Security Number, and date of birth into the person info table. The ethnicity column was deleted from the person info table because it was not needed we only needed either race or ethnicity not both. The resident id was also removed from the person info table because a person has to register and make an account before we get there residential information and moved citizen number into the resident info table in order to combine and keep a match of where the registered user's current address is located. We also removed voters id from the person info for the same reason as the resident id and used the citizen number in the voters table instead. A trigger was created that automatically inserts a user into the voter table after they are placed into the person info table. The trigger gives an auto increment value for their voter id, concatenates their first and last name as full name, then sets if voted equal to 0(false), because they have not voted yet and pulls citizen number over as well. An event was setup for after each election is over the if voted column will be set back to zero or false. the event is to run every 3 years in august right before the upcoming general election. Once the columns were corrected we then realized the hierarchy of our tables which are:

Tables:

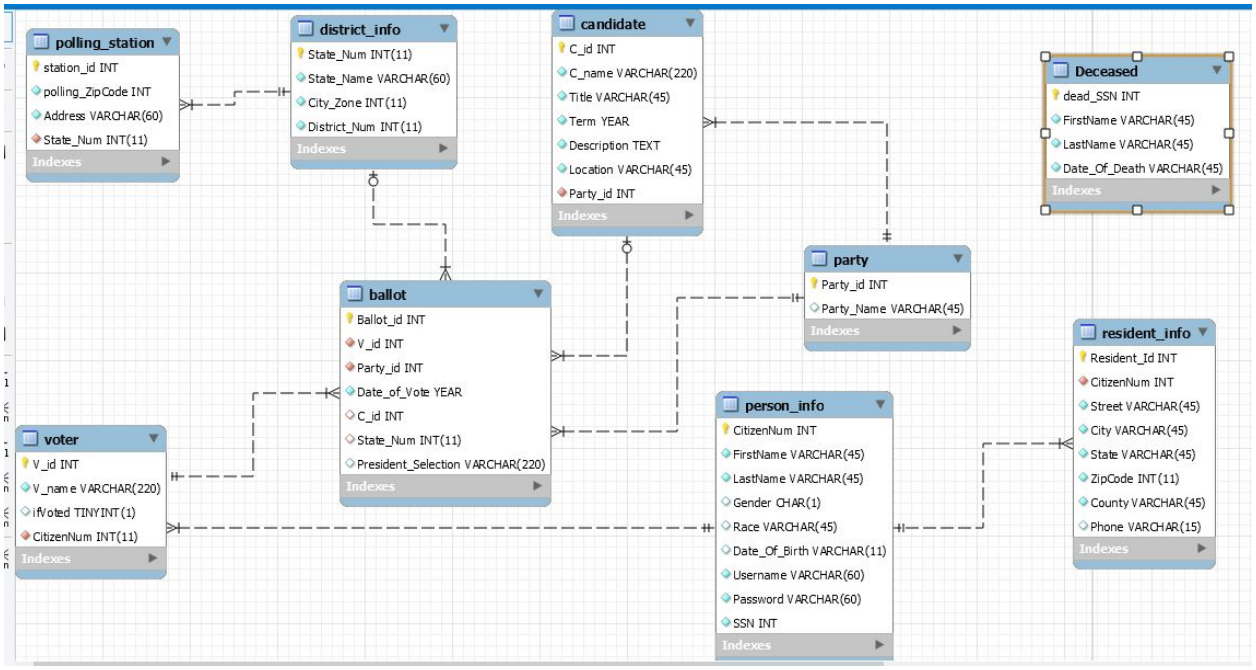
1. Deceased
2. Person Info
3. Resident Info
4. Voter
5. District Info
6. Polling Station
7. Party
8. Candidate
9. Ballot

The design of the district info and polling station table did not go as planned. Problems

came in as we inserted ballots into the ballot table, different people voted for different candidates and it would not show in our database from the way we initially had it setup so we removed the candidate and voter id from the district info table. By removing them from the table it just gives the state name, number, city, and district which is enough to do any analytics we needed. Once the states was established we were going to add the polling station into each state. The candidate and voter id was not needed so it was removed from polling station table also. Next we broke down the address to address and the zip code, then we left the state number because there still needs to be a way to combine the district info with the polling stations.

The party and candidate tables were able to work as we initially planned so we kept them the same. The ballot table did not work as initially designed, since we dropped the facial recognition and the fingerprint the column for disabled voter id and twin id was dropped from the table. The station number had to be changed to state number in which the voter is from for each ballot in the table. The candidate column name changed to president selection to represent to whom the voter actual cast their vote for. The data type for data_of_vote column changed to just the year, because that is all that required for the ballot table.  The column ballot id was kept to allow users to vote in multiple elections. Voter id still remains because there has to be a way to determine who it is that is actually casting their vote. Candidate id, party id and date of vote all remained because they are essential to joining the candidate table and also knowing the year of when the vote is casted. The ballot history table was completely dropped because it was not needed it would just hold redundant information that our ballot table could already hold.

## The conceptual model looks as follows




## Business rules

The business rules for registering to Vote consist of first being a citizen that is at least 18 years or older and must have a valid citizen number. All first time voters' need to create an account that enables them to register and to vote for a presidential election. This process prevent unauthorized user to login and access the system. Once requirement are met, a unique voter id is automatically issued to each registered voter.

The business rules to cast a vote for the presidential candidate is a registered voter can vote for his/her choice for the general presidential candidate online for the given election year by logging in with their username and password. The system only allows the voter to pick only one presidential running candidate of the given election year. After the voter cast his/her vote, it will automatically transfer to the ballot table along with unique ballot id, presidential candidate and year of election.

Insertion, updating and deleting constraints on the database: only new voters are allowed to register to vote by creating a username and password. If the voter is an existing voter, he/she logins with their credential that enables them to vote for the current election. Voter may update their address through the GUI. Only administrators can search user information or delete users, add candidates to the database. If a registered voter passed away, a trigger transfers the voter information to the deceased table. The trigger assigned in the registration table compared the new voter social security number with the deceased voter social security number to verify the user has not used the deceased information to register in the general election.

Once the voter voted for a specific election year, the voter status changes from true to false in the ifVoted column. To change the ifVoter status back to the previous status. We created an event scheduled every three years to roll back the voter status from true to false.

After all the tables were created, the primary keys and the foreign key constraints were established and data are added into each table. To test the triggers on the deceased table we first created 6 deceased people with six different SSN's. Then we created 133 different people to go into the person info table. Each person has their own citizen number, SSN, and username. For each of the 133 people in the person info table we have their residential information which has a unique resident id and it connects to their citizen number. The rest of the information is their address and phone number.  A key thing we needed in the resident info table was the zip code, it is used to determine where the voter will be allowed to cast a vote. In case a user does not have the means to use Electronic Voting System or they require speech monitors/screens that aid with vision impairments the polling stations has the tools.  The next information we inserted for testing was the voter information. We inserted the voter information manually, then

we realized that it could have done updated automatically at the time of registration. Later we went back and created the trigger that will automatically do it. Since we did not want our voter id's to start at one we will manually insert the first voter id at 50001 in then apply the trigger which will insert the rest of the voter's information and auto increment starting with 50002. The party table only has 6 different parties in it: Democratic Party (200), Republican Party (100), Independent Party (300), Libertarian Party (400), Green Party (500), and Constitution Party (600). The party id's start at 100 and increment by 100 for each party. When you get to the general election there is only 3 presidential candidates to vote, so for our database we only have three candidates for each year. In the GUI when a voter goes to cast their vote they will be able to vote for one of the three candidates that is in the candidate table. The insert statement for all the candidates has nine candidates 3 for the 2008 election 3 for 2012 and 3 for the upcoming election in 2016. The district info table has an insert for all 50 including state name, state number (the order they were inserted) a city zone and district zone for each state. The polling stations are still needed in the an electronic voting system, for voters who do not have access to a computer or any electrons to vote, the polling station will have computers for each voter to come in and use the computer to vote. The polling stations include a zip code which is used to match resident's address zip code and tells the voter where to go for their polling station. The polling stations started at 800 and auto increment from there just so all of the numbers did not start at one and get confusing. Every state has at least one polling station, the total number of polling stations is 58. Before we created the GUI for users cast their vote we manually created ballots to go inside the ballots table. By having ballots in the table we could then perform queries to see who won the overall election. How many votes did each person have with the total number of votes at the end. Check the max number of votes for a state and which candidate that was. Count the number of voters for each election. Able to break down

who voted for a particular candidate based on race and or gender.  Once the GUI was created and we created the functionality for votes to cast their votes on the site. The users could cast their vote by entering their voter id and there state number they can make a choice and cast a vote. The GUI has a view that displays a just the states name and state number for the user to see what their state number is.

## Summary of Implemented use cases:

**User Functionality**

-First time user who want to register to vote

Objective: to capture the user personal and demographic information to verify whether the user is qualified to vote.

-Updating existing user information

Objective: A voter changes their address if they move. The user is given the privilege to update their address information. This process helps the database administrator to have up-to-date data about the user.

**Admin User Functionality**

-Administrator dropping a voter from the database

Objective: If the registered voter has voting rights revoked due to felony or some other restriction by the court. The admin user has the right to delete the voter information from the database. By doing this the database store information of voters who have the right to vote.

-Add a candidate for the primary presidential election

Objective: It’s through only the admin user a candidate of any party are allowed to register as a candidate for any presidential election. This process easily identifies the candidate and their party affiliation. By doing so, the admin user can control how many candidate are registered to run for the given election year.

-Looking up the data analysis

Objective: In order to provide accurate data analysis for any interested party of the given election year, all the request for the data analysis can be accessed by the normal user or administrator. By doing so, user can see who won and the breakdown of what type of people voted for a candidate. Admin user can view what user voted for which party.

## *Illustration of Functionality*

For a user who has an account and wants to go in and cast their vote for the election.

1. User goes to the login1.php (evoting login website) and logs in with a user account that is in the database. username = jumpman password = 6rings and clicks submit.
2. From the homepage user clickes find state to find out his state number for the ballot hits the back button to get to the homepage
3. Click the vote now button. Read on which candidate you want to vote for. Find the candidate hillary clinton read her description and what part she is apart of.
4. Scroll down to cast your vote click on the candidate you want then enter your voter id (50052) and the state number (10). Click submit and it will take you to a page which says you successfully casted your vote.

For a person who never voted and wants to register to vote

1. Go to the login1.php page (evoting homepage) click log in with not credentials then click the register now link that comes up.
2. Insert all the information that comes up First Name: david  Last Name: wilson  Date of Birth: 1955-12-10 Phone Number : 7043781000 Social Security Number:  046695895 Citizen Number:7045108800 Gender: M Race: White Street: 9 east university street City: CENTRAL County: meek State: AK Zip Code: 99730 Username: dwilson Password: dba6120. then click submit.
3. Go to the home page and click account info to get your voter id (50074) to see that you are a registered voter with a valid voter id. click back
4. Click find polling station.
    - Enter 99730 to see where the address of the closest polling station
        i. Feedback will say STEESE HIGHWAY, CENTRAL, AK.

User who wants to login and wants to change their address.

1. log in to the system with the credentials username = jumpman password = 6rings and clicks submit.
2. click account info then input 830111129 the enter street:239 akins way city: charlotte state: NC county: mecklenburg  zip:28216 phone:7043458000 and click submit.
3. Then it will say your residential information has been successfully updated and it reads backstreet:239 akins way city: charlotte  state: NC county: mecklenburg  zip:28216 phone:7043458000
4. Click logout to logout the system.

User who wants to check the analytics for the previous election.

1. Log in to the system with the credentials username = jumpman password = 6rings
2. Click election stats

    - Takes you to who won the popular vote then click back
3. Click post presidential stats
    - Feedback gives you presidential statistical breakdown per presidential candidates then click back click state stats
4. Click state stats
    - Feedback gives you statistical breakdown of percentages  per state click back
5. Click state party analysis
    - Feedback ives you breakdown the statistical breakdown per state that determines party based on the majority vote for 2012

Administrator wants to retrieve a user's username and password to help them log in.

1. Log in with username: Admin187 password: password187
2. Click admin control
3. Insert the number of sql row to retrieve from the database. insert 70
4. The user will then tell you their first name: stephen, last name: curry, last four of SSN: 7382, zip code: 95249 and birth year 1982.
5. Then we would read stephen curry his username and password which is username: allday and password: awerty

Administrator wants to delete a user because they are no longer allowed to vote.

1. Log in with username: Admin187 password: password187
2. Click delete user button
3. Input the citizen number: 829630011, voter id: 50051, and user's  full name: Lebron James click submit.

- Feedback that tells you, you deleted voter Lebron James with citizen number: 829630011 and  voter id: 50051 from the person info, resident info, and voter table.

Administrator wants to add candidate for the upcoming election.

1. Login with username: Admin187 password: password187
2. Insert candidate name: Yi Zhen, term: 2020,  Description: vote for me! location: Charlotte and party id: 200 then click submit
    - Feedback that tells you you inserted candidate name: Yi Zhen, term: 2020, Description: Vote for me! location: Charlotte and party id: 200.

Administrator wants to look at voter analytics

1. Login with username: Admin187 password: password187
2. Click on current result
3. Click either democratic or republican
    - Feedback that shows a list of voters and the candidate they voted for, either democratic or republican

## Summary Discussion

In conclusion the database combined with the GUI allows users to register to vote if they meet the criteria. then a user is able up update their resident information if they move, as many people move for different reasons. Then the users can look up information about the candidates that are running, their state name and number, and where they are allow to go cast a vote if they need a computer. The user can look up there specific polling station by entering in there zipcode. Users can look up analytics from the previous election. they can see who won, how many people voted, and a breakdown of how certain groups of people voted. There are a few

issues we did not have time to cover and still wanted to add. The fingerprint and facial recognition functionality was something we thought would be extremely useful and new since a lot of technologies today are using them. Cell phones are big with fingerprint scanners. Facial recognition is used in the apps like snapchat and facebook. Unfortunately we lack the time and software to actually implement these two features.  We wanted to finish the main functionalities first then if timepremitied we were going to add them in but time did not permit. We also want the voter to get a final view of their selection before submitting it. We were able to print out the the choice but because we could not print it out on another page we thought it would be safer to just move the voter to a new page so they will not cast multiple votes. If given more time we would figure how how to let the voter know which candidate they voted and also move them to a separate page at the same time. We had to learn how to code with php and if we had more knowledge of it would would of been able to code a session to hold all the way through and then the voter would not have to enter in the there voter id when they cast their vote or type there citizen number when they go up update their address it would just work for the voter that is logged in the system. We wanted to make a trigger that updates the if voted column in voter table to true once they vote. The trigger was created successfully but it did not work in front end and we ran out of time on the project. Lastly we realized that the deceased table will get really extremely large after a while and may cause issues and make the system run slower. A temporary fix for the issue would be to move the data from the deceased table to an external harddrive to create space for the database.

The most challenging part of the project was learning how to code in html, php, and javascript. The group as a whole did not have knowledge of php before we started. We read murach's php/mysql book watch youtube videos and look at other resources online to get an understanding of what we needed to do to get what we wanted. We also wanted to add graphics

to the pages to give a visual display so we had to learn javascript and use google to make the graphs. We received outside help from another students who explained how it works and showed us where we can get extra help. Figuring out the foreign key constraints, determining which tables needed which was challenging. We figured it out by using logic and ordering the tables as how the order should be for one person to become a user and vote.

**Breakdown**

| **Shemeles** | **Dion** | **Ikran** |
|---|---|---|
| Triggers/events | Html view | Php on the Html Pages |
| Views/ Queries | Create database script | ER Diagram |
| User Authentication | Create Sample Data | Javascript for analytics |
| install xampp | install xampp | install xampp |

We divided the work in this manner because we felt this was the best way to work with our strong areas. Even though we did not have much experience, we still dived it to whoever wanted to take on the task and learn. We did not just make that one person do all the work for the task we each helped out and gave input but they were the task leader.