Jonathan Hanobik and Spencer Cousino

Software Plan - Exercise 4

CS1631, Spring 2017

16 February 2017

1.0 Scope

 For our product, we will be designing a mobile voting application that will allow voters to cast their ballot through text. Running on an Android device, our application will receive the text, parse it for information, and update several tables. When the voting is complete, the administrator will end the session and display the results. We hope to have this project complete for CS Day 2017, so that it can be utilized during the poster competition.

1.1 Functions

 This product will have several functions. Firstly, it will be able to **accept a vote**. Upon accepting a vote, the application should **decode it** to be sure that it is of the correct format. Upon validation of the vote, the product will **update both a tally and voter table within a database** to achieve consistency among the voting period. Once the voting is over, the product must **display the results** and show the winner. Each of these will be discussed further below:

● Accepting a Vote - when the program is accepting a vote, it will be waiting and listening for text messages sent to the phone number of the device in which it is running. Each and every vote will be noticed, but not all of them will be appropriate for our application’s purposes. At this point, this function cannot know which messages are valid or not. It simply takes the incoming message and passes it on. Also, at this stage of the design process, only text message votes will be accepting. Any other form of voting will not be included in the results of this program.

● Validating a Vote - after the vote is received, it must be validated to know that it is appropriate for this application. This function will check to make sure that only a single identifier (in our case, a numerical value) was sent and is within the range of the possible candidates. If that is correct, this method will also check the phone number associated with the vote to be sure that this voter hasn’t cast a ballot before. Using the database, the phone number will be checked. If it is not listed, the vote is good and will be cast. If, upon checking the database, the phone number is located, the vote will not be cast. This method will not be updating any database, only querying for the information it needs.

● Updating the Database Tables - upon knowing that the vote is valid, the product must update the tally table in the database for the candidate who received the vote. Additionally, the program must enter the phone number of the voter into a different table in the database. This will prevent the same voter from casting a ballot again in the future. The function that utilizes this information is the one that validates the votes. This is a separate method that deals mainly with updating the tables of the database to a consistent and correct state.

● Display the Results - when the voting period is over and the administrator decides to cease any more ballots from being cast, the product will take steps to finalize the procedure. The administrator will prevent anymore votes from being cast, and the product will display the name and number of votes from the winning candidate. Besides displaying the results, this function cannot alter or do anything to the state of the application.

1.2 Performance

 This product should be able to handle many users from CS Day who would like to vote. Whether there are 10 or 100 voters, the application should update the database so that it is in a consistent and durable state. In short, it is important that the application handles any volume of potential voters and keeps everything in a correct and consistent state.

1.3 Limitations

 As far as limitations, this product cannot support voters voting from either their phone or email. Because our application will hold a database of either phone numbers or e-mail addresses (another version of the product), it cannot determine which e-mail address corresponds with which phone number. This would create a consistency issue and would not be able to prevent users from casting multiple votes from different devices.

 Additionally, our application does not allow users to retract their vote and cast a new one. Once a vote is sent from a phone number, it must remain in the database. Although this is something that we could consider in future releases of our application, it is not something we can support at this time.

2.0 Tasks

 From our viewpoint as developers, there are a few tasks that must be completed for our voting application. Firstly, we must develop the user interface, which will allow voters to cast their ballot. It is worth noting that no GUI will be presented to the user at this stage. The voter will simply send their vote. This is the only task through the user interface, and it must be made clear to the consumer how to communicate with our application. On the developer side of the user interface, the votes would need to be validated - Does the message contain a single digit representing the preferred candidate? Is this voter casting their ballot for the first time?

 Aside from that, we have discussed having a graphical user interface for the application administrator. This would enable the administrator to begin and end voting as they like. This will be a very simple graphical user interface, with perhaps a button or two for initiating or ending the voting.

 Finally, a major task in our application will deal with managing the voter and tally tables in the database. Upon receiving a vote via text and parsing it to be sure that it is valid, the application would need to update the table containing the voters who have cast their vote and the table of the tallies for each candidate. The voter table would require an “INSERT” query, while the tally table would require a modification of a count. Both are relatively simple, but need to keep our application in a consistent state.

3.0 Resources

3.1 Hardware

 For this application, the hardware required would be a computer to develop the device, a mobile application to run the application(administrator), and one or more mobile devices for the consumers to utilize the application. After designing, writing, and debugging the application on a computer, it would then be uploaded and run on the administrator’s Android device. After that, each user who wishes to use the application will simply send a text message to a provided phone number with the candidate identification number of their choice.

3.2 Software

 As far as software, our application would require Android Studio for development. There are so many functionalities contained within Android Studio, so most of our database and message handling abilities will be taken care of. We would also need the correct APIs for our project, so that we could run on a specific build platform. Additionally, we will be utilizing the SIS server and remote for testing and message passing between our components.

3.3 People

 There will be very few people involved in our application. Besides the obvious developers needed, our application only requires one person to administer the voting (deciding when to begin or end the voting). Aside from him/her, the application would just need users to cast their votes and select which candidate they prefer.

Planning Questionnaire for Mobile phone voting project

**UML Diagrams**

Use Case Diagram:



Class Diagram:



Activity Diagram:



Collaboration Diagram:



Deployment Diagram:



**Questionnaire**

To help you develop a software plan so that we can have a meaningful discussion when we meet, please complete the following questionnaire by noon time before meeting and e-mail to me, and bring hard copies when we meet. There will be a two-point penalty to each group member if the planning questionnaire is not submitted to me on time.

Date: February 16, 2017

Group ID: 3

Member #1 (last name): **Hanobik**

Member #2 (last name): **Cousino**

1. Input mode. Please select one:

**1.1 (X) Voter will send a short text message from their mobile phone to my smart phone (Note: don't select this option if your phone is not a smart phone, or there is no way to access the incoming short text message by a phone-resident input processor, or your phone is not 100% available on CS-Day.)**

1.2 ( ) Voter will send a e-mail from their mobile phone to my smart phone (Note: don't select this option if your phone is not a smart phone, or there is no way to access the incoming e-mail by a phone-resident input processor, or your phone is not 100% available on CS-Day.)

1.3 ( ) Voter will send a e-mail from their mobile phone or computer account to my computer account. (Note: Do not select this option if you cannot access your e-mail from your notebook or PC, or there is no way to access the incoming e-mail by an input processor running on your notebook or PC, or your notebook or PC is not 100% available on CS-Day.)

1.4 ( ) My group does not have any of the above. (In which case I will give you special permission to withdraw.)

2. If you select 1.1 or 1.2, answer the following questions:

**2.1 My smart phone's phone number: \_724-610-7369\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

 **(Voters will send text message/e-mail to this number which will be printed on the poster on CS-Day)**

**2.2 My smart phone manufacturer, model number and other hardware details:**

 **\_Pixel XL Manufactured by HTC/Google Running 7.1.1\_\_Model No: G-2PW2100**

**2.3 (X) My smart phone runs Android and I know how to develop an Android app as the input processor and another Android app as the VotingSoftware component**

2.4 ( ) My smart phone does not run Android but I know how to develop an app as the input processor and another app as the VotingSoftware component in the following language:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.5 ( ) My development platform is my smart phone (Note: This is not highly recommended)

2.6 **(X) My development platform is my notebook or PC running under:\_\_Windows + Android Studio\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and I will be able to port the app to my smart phone.**

3. If you select 1.3, answer the following questions: (Note: I expect most groups will select this option, however bonus points will be awarded for groups selecting either option 1.1 or option 1.2.)

3.1 My computer account to be used to receive incoming

 e-mail votes: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2 Hardware details about my notebook or PC:

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3 Software details about my notebook or PC:

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Development Diagram:

**4.1 (X) With the above information I think I know how to draw a development diagram**

4.2 ( ) With the above information I still have no clue

 how to draw a development diagram. My question is:

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Deployment Diagram:

**5.1 (X) With the above information I think I know how to draw a deployment diagram**

5.2 ( ) With the above information I still have no clue

 how to draw a deployment diagram. My question is:

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_