Requirements Specification

Dream Team: Tiffany Huynh, Sam Smith, Roddrick Henderson

I. Introduction

- **Purpose of the system**
  - The objective of this project is to produce a universal voting platform for students.
- **Scope, objectives, and success criteria of the system**
  - It will consist of supporting various contests consisting of multiple candidates. JSVS will allow students to register to vote through a secure system that limits ineligible voters and ballots.
- **Definitions, acronyms, and abbreviations**
  - JSVS - Jaguar Secure Voting System
- **References**
  - Stakeholders, student body, faculty members, the public community
- **Overview**
  - To improve upon the current voting system in place at the University of South Alabama.

II. Current System

- Student body is currently utilizing an email method through google poll as a voting system.

III. Proposed System

- **Overview**
  - Our goal is to create a functional app that will allow students to cast their votes easily on a handheld device.
- **Functional Requirements**
  - The system will allow users to be able to log with their JagID and password.
  - The system will then search for the registered user in the admin database for existing accounts with given credentials.
  - The system will authenticate and verify voter verification status.
  - The system will display a list of candidates for the voter to view and select from.
  - Admins will be able to add polls with desired candidates and submit polls for voters to choose from.
  - Once the user presses the confirm button, the system must be able to confirm successful ballot submissions.

- ○ Admins will be able to select which polls will be active and to end active polls.

Legal -

1. Should this voting system be exclusively only licensed to USA associates?
   - ● Yes

Interface -

Where will our inputs/outputs come from?

- ○ The system's we'll be interfacing with will include the database for storing registrant information, as well as candidate and ballot results. We'll also need to interact with the voter registration system for registrant validation.
- ○ Our outputs will be the users' mobile devices and the previously mentioned databases.

- **Nonfunctional Requirements**

Usability -

1. What will the voting system interface look like?
   - ● Adaptable to different mobile devices and easily readable
2. How will voters access this system?
   - ● Via handheld app or voting station
3. Who is this system made for, will it only be exclusive to that specific group?
   - ● USA students, faculty, and staff, with the focus on students.
4. What exactly do you want to achieve in this system?
   - ● Implementation of a complete, fully documented voting system
5. How many people will be featured in the polling at maximum?
   - ● ~20k

Reliability (robustness, security, safety) -

1. Why are you looking to change how you are currently voting now?
   - ● To increase participation and voter confidence in outcomes and privacy.
2. How will user data and inputs be treated?
   - ● Every user's login and votes will be kept private from other users.

Performance -

1. How long does the system take to respond to user input?

- Optimized for most devices to smoothly operate the system

Supportability -
1. In what form of method do you want updates and bugs resolved?
    - We want the user to be able to easily retrieve updates from the source (our team) and have the option between automatic updates and manual updates.
    - We will have to make sure this is integrated between different application management systems such as the IOS AppStore, Google's PlayStore, etc.
2. In what way are we supporting international students?
    - Make sure the information is easily digestible for any type of persons
    - Make sure we support multiple languages.

Implementation -
1. What platform will this system operate on?
    - Via handheld app (Probably only APK or Google Play Store at first)

Operation -
1. What are the constraints that this voting system will perform within (people, environments, technology, etc..)?
    - Stakeholders: USA network, Computer Center, student government, faculty government, Deans, Chairs, VPs

Packaging -
1. How should a user receive our application?
    - A user will either access the application via a web page or a dedicated app that will be delivered via APK (or google play) connected to that web page.

- **System Models**
  - **Scenarios**

5.1 Scenario Name	userLoginDenial
Participating Actors	SamIAM:voter; SamIAM:administrator
Flow of Events	1. SamIAM Attempts to login 2. The system searches the registered voter database/registered admin database for existing accounts with given credentials, but doesn't find a match 3. The system returns to the user a "login failed" dialog box, and warns the user that if they try too many times that the account will be locked 4. SamIAM attempts to login too many times and account is locked 5. The system prompts the user to reset their password

5.2 Scenario Name	userLoginAck
Participating Actors	RODDERICK:voter; RODDERICK:administrator
Flow of Events	1. RODDERICK attempts to login to the application using personal credential=JagID and password 2. The system searches the registered voter database/registered administrator database for existing accounts with given credentials 3. The system locates and verifies an account that matches the given JagID and password 4. The system returns a dialog box stating the success of RODDERICK's login attempt 5. The system displays the poll dashboard

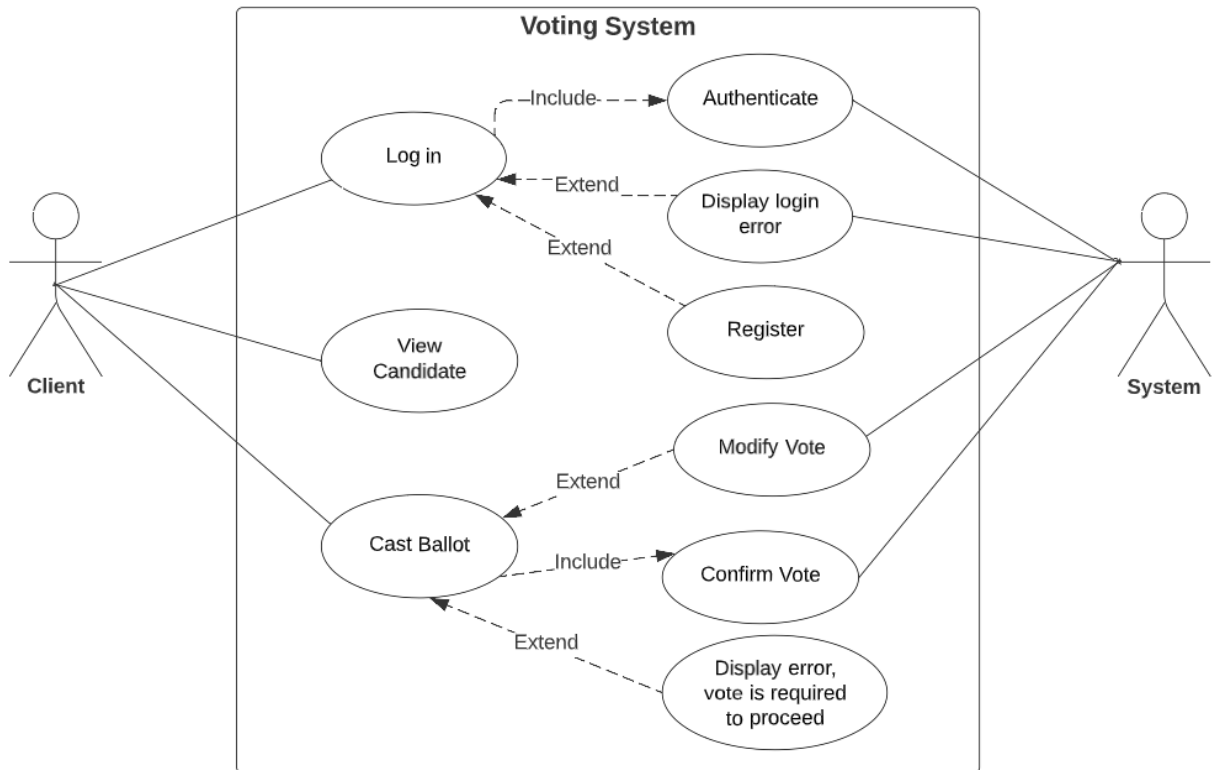| 5.3 Scenario Name	userMultipleVote
Participating Actors	Tiff:voter
Flow of Events	1. Tiff selects more than one candidate on a poll that only allows the voter to select one candidate. 2. Error dialogue message prompts the user to only select one candidate 3. Tiff selects no candidate. 4. Error dialogue message prompts the user to select one candidate before user can proceed to the next screen 5. Tiff finally selects a candidate and the system checks if there is any bad input detected. 6. System acknowledges good input and stores the vote. 7. User is shown the success screen that vote was counted for.

5.4 Scenario Name	userRegister
Participating Actors	Bert:voter
Flow of Events	1. Bert is not currently registered in the database, he either had this knowledge beforehand or found out when he attempted to login (scenario 5.1) 2. Bert chooses the register option on the login screen 3. Bert inputs the correct information into their respective fields 4. The system responds by telling Bert whether or not his registration process was successful 5. The system displays the login page again

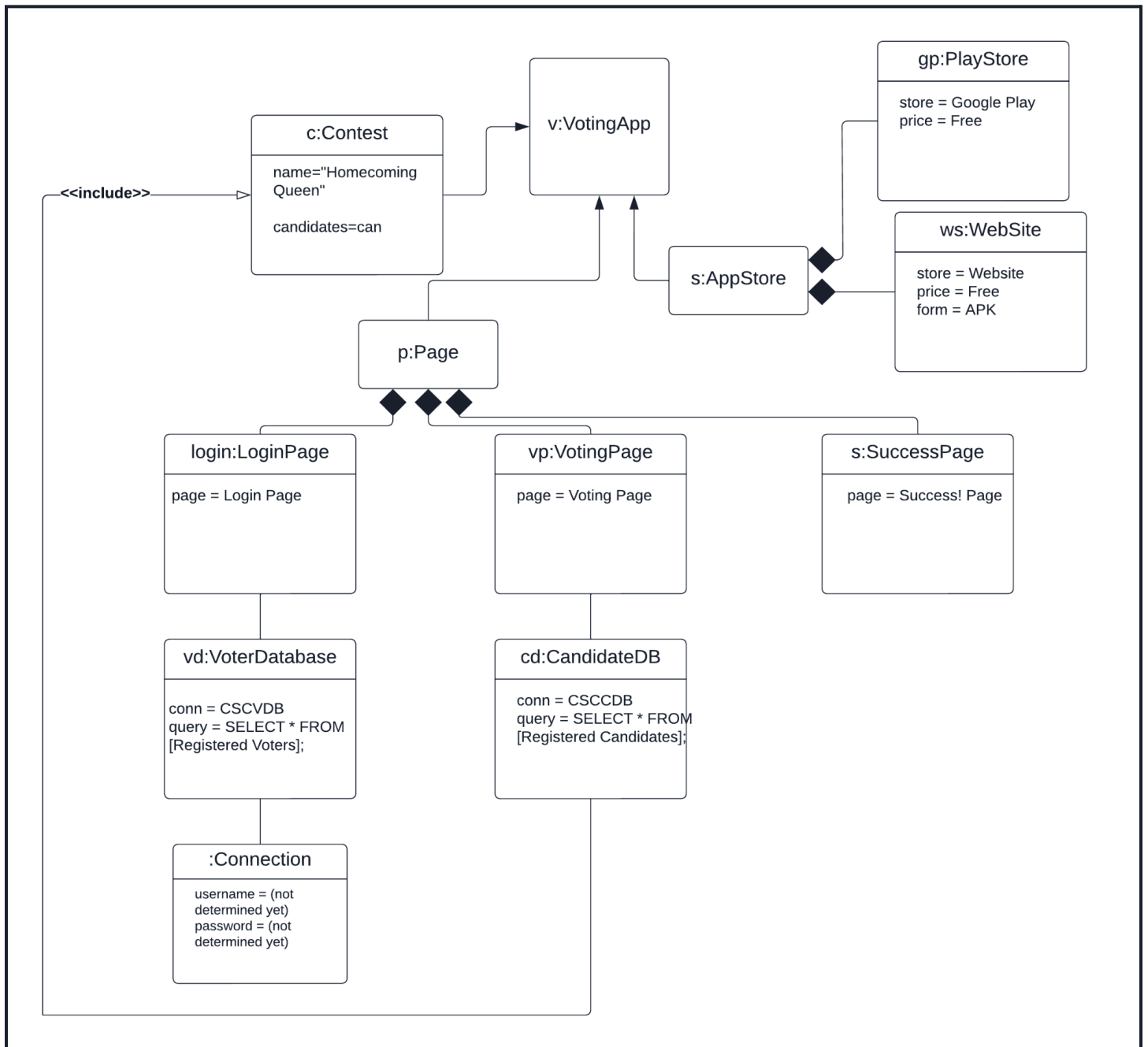| 5.5 Scenario Name	adminLoginFailure
Participating Actors	Jane:Administrator
Flow of Events	1. Administrator attempts to log in. 2. Administrator does not have the credentials to access voting application panel. (scenario 5.1) 3. Error dialogue message prompts administrator to request access to voting application panel through ticket submission. 4. Administrator is notified via email that access has been granted.

5.6 Scenario Name	adminAddPoll
Participating Actors	Jane:Administrator
Flow of Events	1. Administrator successfully logs in (scenario 5.2) 2. Administrator is presented with an administration panel 3. Administrator selects the "Add a Poll" option 4. Administrator selects the poll from the list of polls available in the election database 5. The system checks for election in database, verifies its existence 6. Administrator is presented with admin panel including newly added

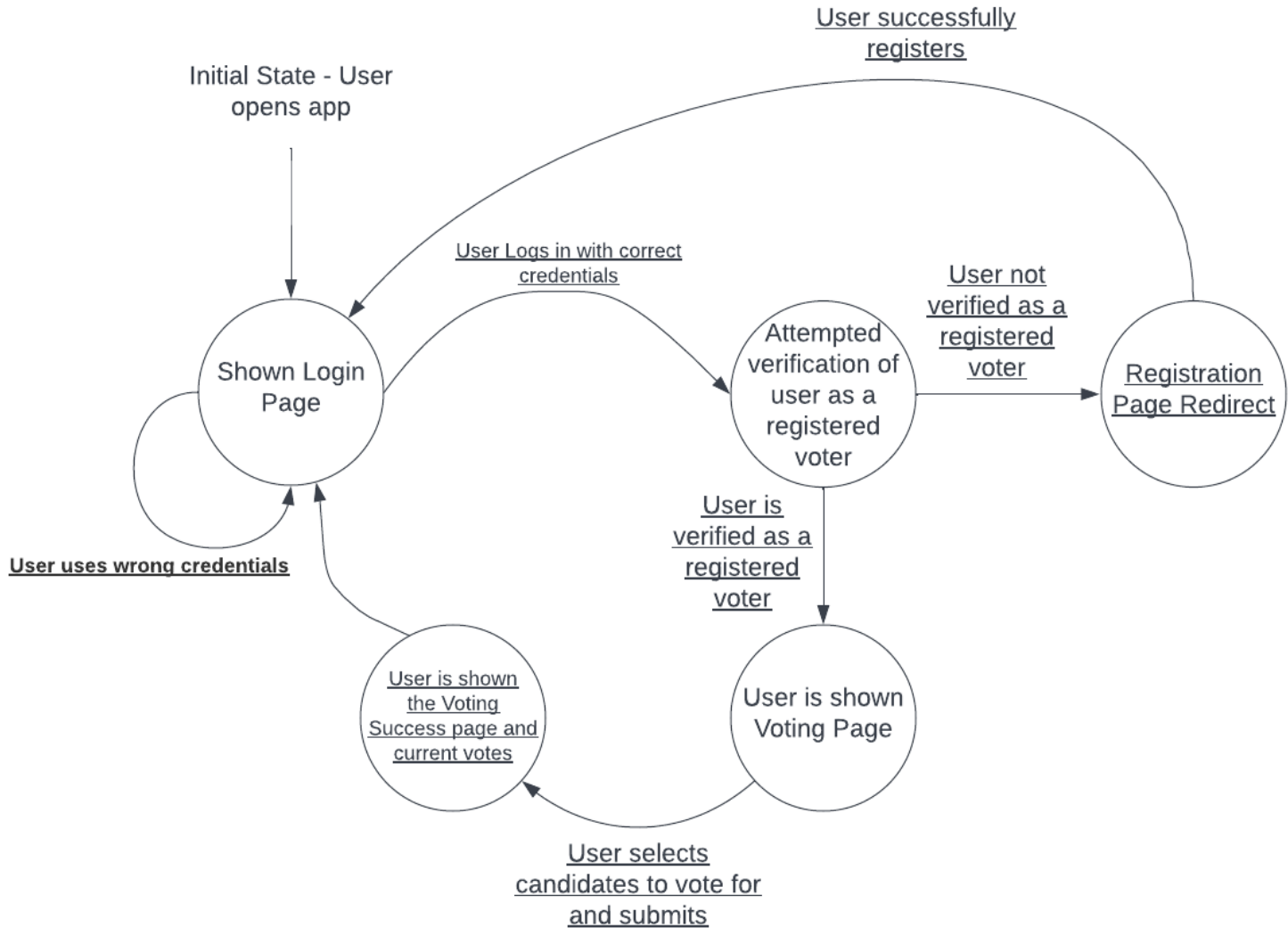| 5.7 Scenario Name	adminRemovePoll
Participating Actors	Jane:Admin
Flow of Events	1. Administrator successfully logs in. (scenario 5.2) 2. Administrator is presented with an administration panel. 3. Administrator selects the "Remove Poll" option 4. Message dialogue prompt is displayed on the screen to confirm that the user wants to remove the poll. 5. System removes the poll/election from the Voter;s options

○ **Use Case Models**

○ **Object Model**

gp:PlayStore

store = Google Play
price = Free

c:Contest

name="Homecoming Queen"

candidates=can

v:VotingApp

<<include>>

ws:WebSite

store = Website
price = Free
form = APK

s:AppStore

p:Page

login:LoginPage

page = Login Page

vp:VotingPage

page = Voting Page

s:SuccessPage

page = Success! Page

vd:VoterDatabase

conn = CSCVDB
query = SELECT * FROM
[Registered Voters];

cd:CandidateDB

conn = CSCCDB
query = SELECT * FROM
[Registered Candidates];

:Connection

username = (not
determined yet)
password = (not
determined yet)

○ **Dynamic Model**

Initial State - User opens app

Shown Login Page

User uses wrong credentials

User Logs in with correct credentials

Attempted verification of user as a registered voter

User successfully registers

User not verified as a registered voter

Registration Page Redirect

User is verified as a registered voter

User is shown Voting Page

User selects candidates to vote for and submits

User is shown the Voting Success page and current votes

- User Interface - registration/login screen, ballot screen, successful submission screen

Welcome
to
Jaguar Secure Voting System!

REGISTER

RETURNING USER

REGISTER

FIRST NAME

LAST NAME

JAG ID

MAJOR

LOGIN

JAG ID

PASSWORD

☐ REMEMBER ME

☐ FORGOT PASSWORD?

CANDIDATES

PLEASE SELECT ONE OF THE OPTIONS BELOW:

☐ CANDIDATE #1

☐ CANDIDATE #2

☐ CANDIDATE #3

☐ CANDIDATE #4

☐ CANDIDATE #5

SUBMIT

Your vote was submitted.

✓

	Polls	Last Modified	Action
	+ Add a poll		▽
☐	**Polls**	**Last Modified**	**Action**
☐	Homecoming Queen	October 18, 2022 24:10:00	✏ 🗄
☐	Homecoming King	October 20, 2022 13:35:00	✏ 🗄

<< Previous   Next >>

Submit