**Software Requirements and Design Document**

**For**

**Group <ResistorCalculator>**

Version 2.0

**Authors**:

Jorge Atencio

Jacob Hobson

Schmidt Jean

Ronald Nazaire

Vinzce Yadao

# Overview (5 points)

The application is a Resistor Calculator targeted towards FSU/FAMU College of Engineering Students/Faculty. We have shopped the idea around with several FSU/FAMU students and professors, they believe that this would be a very helpful tool to use during their class and labs. The tool calculates the ohm value and tolerance based on the resistor color codes, the total resistance of a group of resistors in parallel or in series, and the resistance of a conductor based on size and conductivity. The calculator features the support of multi-band calculations (4-6 bands). The backend of the application supports user accounts to back up to a database for creating saved resistor values for future use. The front end will implement a cohesive and easy-to-understand UI that matches material design standards.

# Functional Requirements (10 points)

**0 = Done**

**1 = High Priority**

**2 = Medium Priority**

**3 = Low Priority**

**1) Resistor Calculation**: Decodes the information for color banded axial lead resistors. Able to select the number of bands, then the colors of the bands. Then calculates the value and tolerance of the resistors

**1) Multi-Band Implementation:** Support of 4, 5, and 6 band resistors. Full Resistor Calculation functionality for multiple band values.

**2) Updating the Resistor Visually**: Should be able to see a visual representation of the Resistor when changing each band color.

**2) Saved Resistor Values**: Able to save resistor values with a picture of the saved resistor

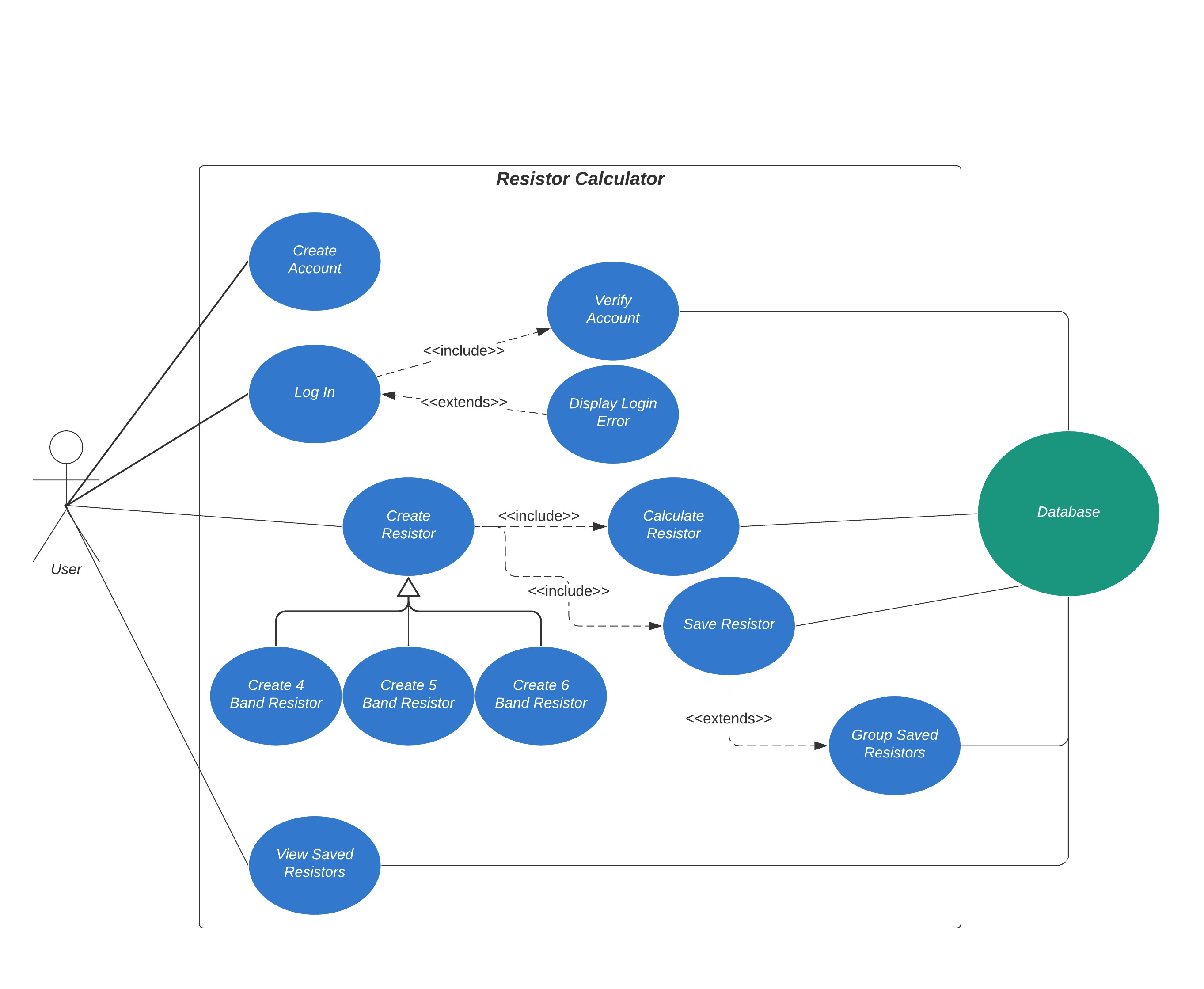
**3) User Log In**: Users are able log in. Logging in allows a user to save resistor values and the visual representation of the resistor for a later use.

**3) Grouping of Saved Resistor Values***:* Able to store saved resistor values under a group. Example: Saving the values of 4 different resistors under a group named “Project”.

# Non-functional Requirements (10 points)

* In the login page, three failed password attempts will prompt a security question
* Passwords should never be viewable at the point of entry or any other time
* Display should be easily understandable with clear font and without truncating displayed text
* Program should be able to process calculations at a high rate of speed
* User calculations should be reliably stored under their account should they chose to save it
* All resistor calculations must be accurate to two decimal places
* Program will allow up to 5 stored calculations per user
* Cohesive and Understandable UI design
* A stable non-crashing application
* Ability to run on any web browser

# Use Case Diagram (10 points)



1. Name: Create Account

Participating actors: User

Entry condition: User clicks the Create Account button

Exit condition: User clicks the back button

Flow of events:

1. User opens the web page

2. User is on the login page

3. User clicks on create account button

4. User inputs account information

2. Name: Log In

Participating actors: User, DataBase

Entry condition: User enters the webpage

Exit condition:

User clicks the back button

Flow of events:

User opens webpage

User is on login page

User inputs log in information

User clicks the login button

3. Name: Create Resistor

Participating actors: User, Database

3. Entry condition:

User clicks the Create Resistor Button

4. Exit condition:

User clicks the back button

5. Flow of events:

User clicks the Create Resistor button

User selects bands

Database updates the resistor value as bands are selected

4. Name: Save Resistor

Participating actors: User, Database

3. Entry condition:

User clicks the Save Resistor button

4. Exit condition:

User clicks the back button

5. Flow of events:

User clicks the Save Resistor button

Database saves the image of the saved resistor

Database saves the value of the saved resistor

5. Name: View saved resistor

Participating actors: User, Database

3. Entry condition:

User clicks the View Saved Resistor button

4. Exit condition:

User clicks the back button

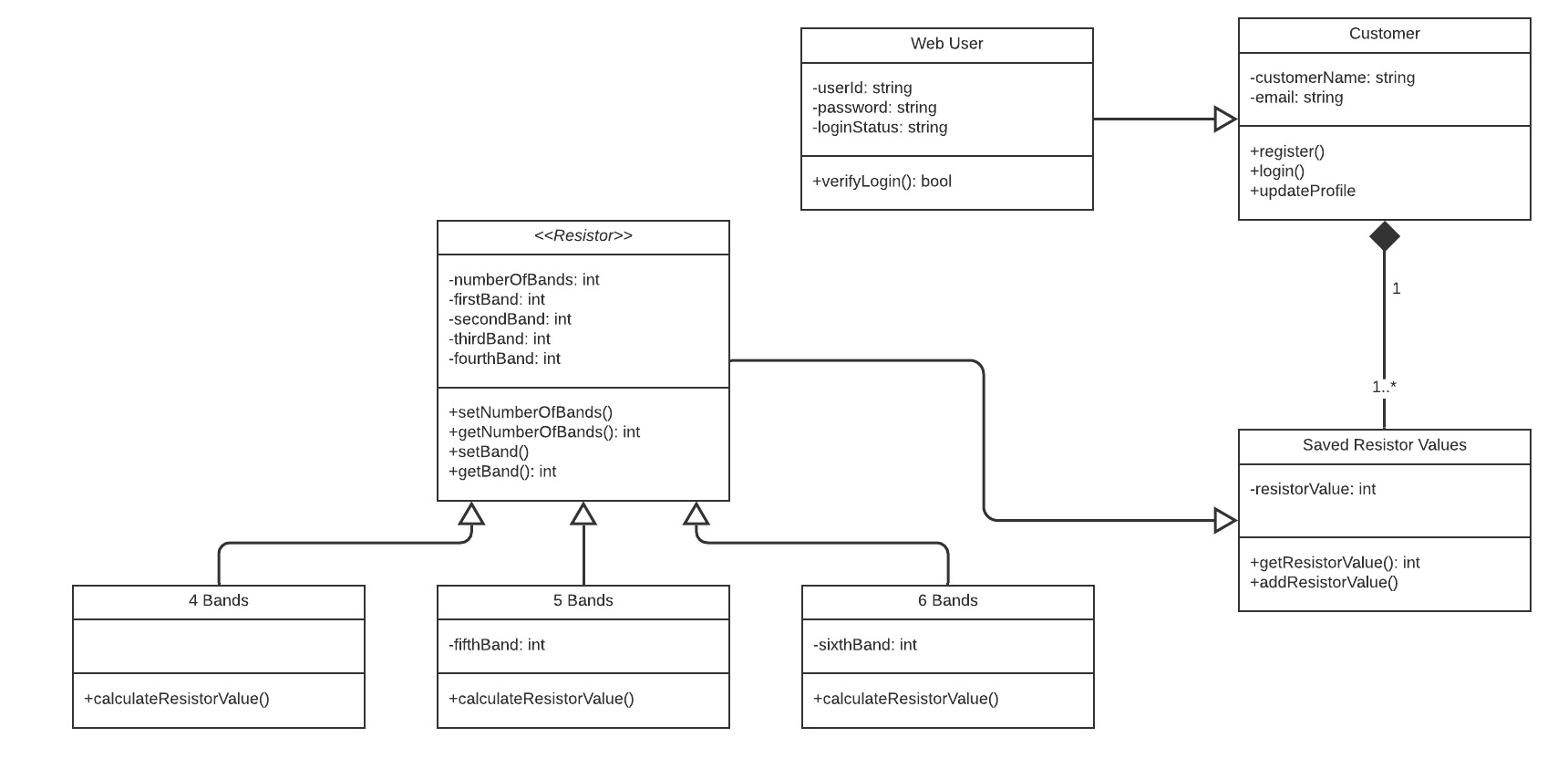
5. Flow of events:

User clicks the View Saved Resistor button

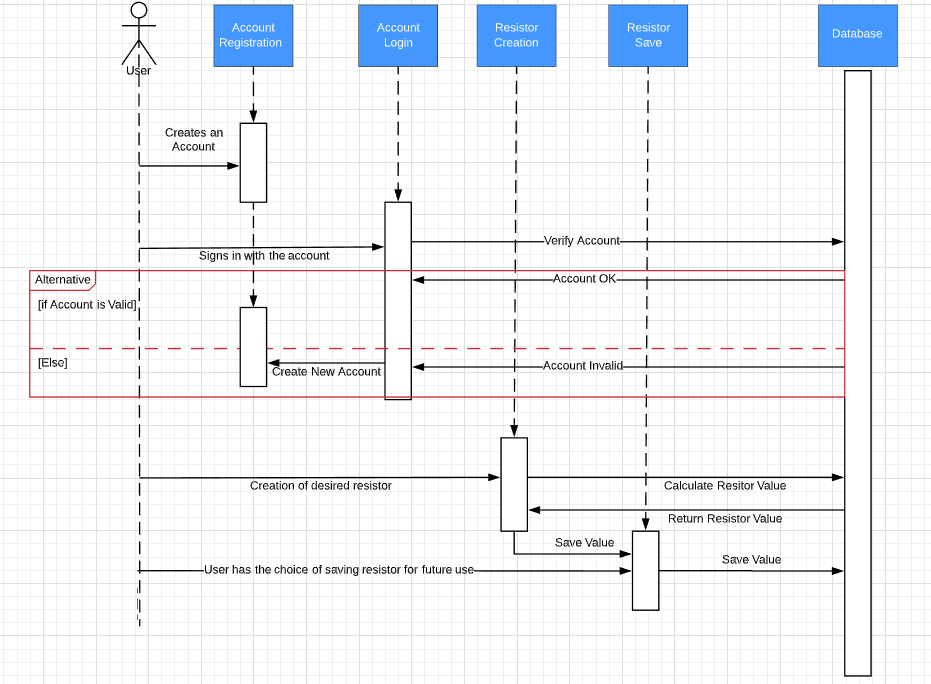
Database pulls up the images and values of saved resistors

# Class Diagram and Sequence Diagrams (15 points)

Class Diagram:



Sequence Diagram:



# Operating Environment (5 points)

Any web browser that has an internet connection and a valid google account

# Assumptions and Dependencies (5 points)

The application is depending on a SQL server in order for a user to login and register their account. The SQL server is also dependent on storing the saved resistor values.