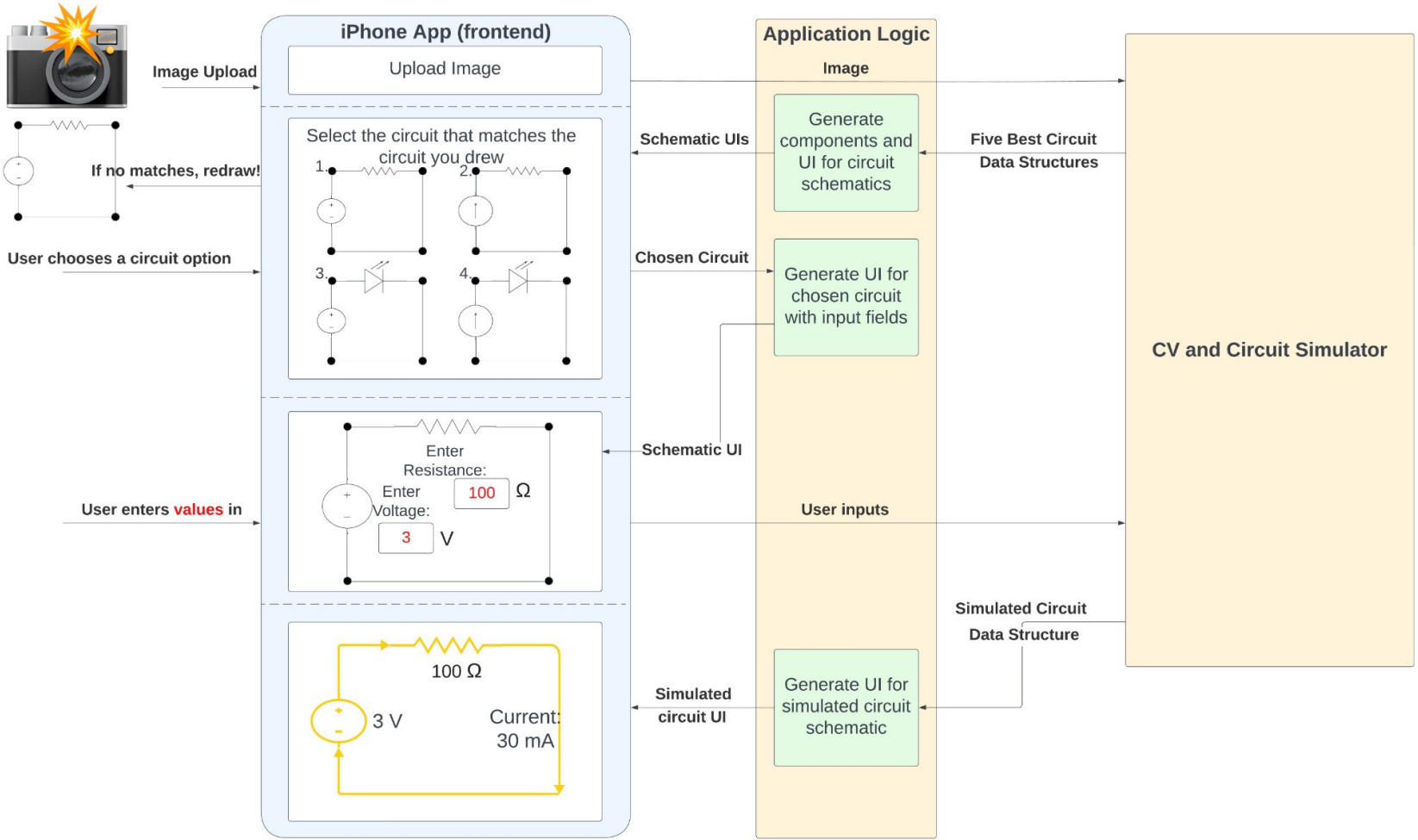
Use Case Revisited

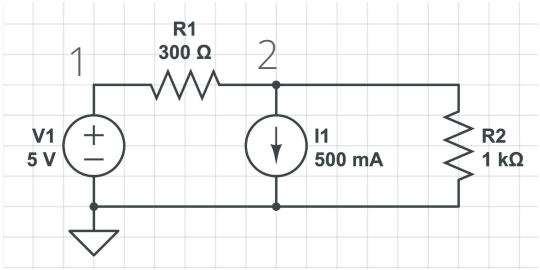
- Goal: Provide a free application to help middle+high school students learn how DC circuits work
- Usability
  - Accessibility: application is **lightweight: free of cost, <= 100MB**
  - **80% approval rate** from test group
- Individual Component Detection Accuracy
  - **90% detection accuracy** from unit tests
- Combined Component Detection Accuracy
  - Display **correct circuit 90%** of the time
- Simulator Accuracy
  - **100% correctness** on analyzing given circuit


iPhone App (frontend)
Application Logic
CV and Circuit Simulator
Image Upload
Upload Image
Image
If no matches, redraw!
Select the circuit that matches the circuit you drew
1.
2.
3.
4.
Schematic UIs
Generate components and UI for circuit schematics
Five Best Circuit Data Structures
User chooses a circuit option
Chosen Circuit
Generate UI for chosen circuit with input fields
Enter Resistance:
Enter Voltage:
100
Ω
3
V
Schematic UI
User enters values in
User inputs
Simulated Circuit Data Structure
100 Ω
3 V
Current: 30 mA
Simulated circuit UI
Generate UI for simulated circuit schematic

Circuit Simulator - Modified Nodal Analysis

- Receive **netlist** generated by user inputting values after confirming circuit
- Objective-C++ wrapper to cast C++ data to Swift data
- Run simulation by performing **modified nodal analysis**
- Steady state **DC analysis**

```
V1 1 0 5
R1 1 2 300
I1 2 0 0.5
R2 2 0 1000
```

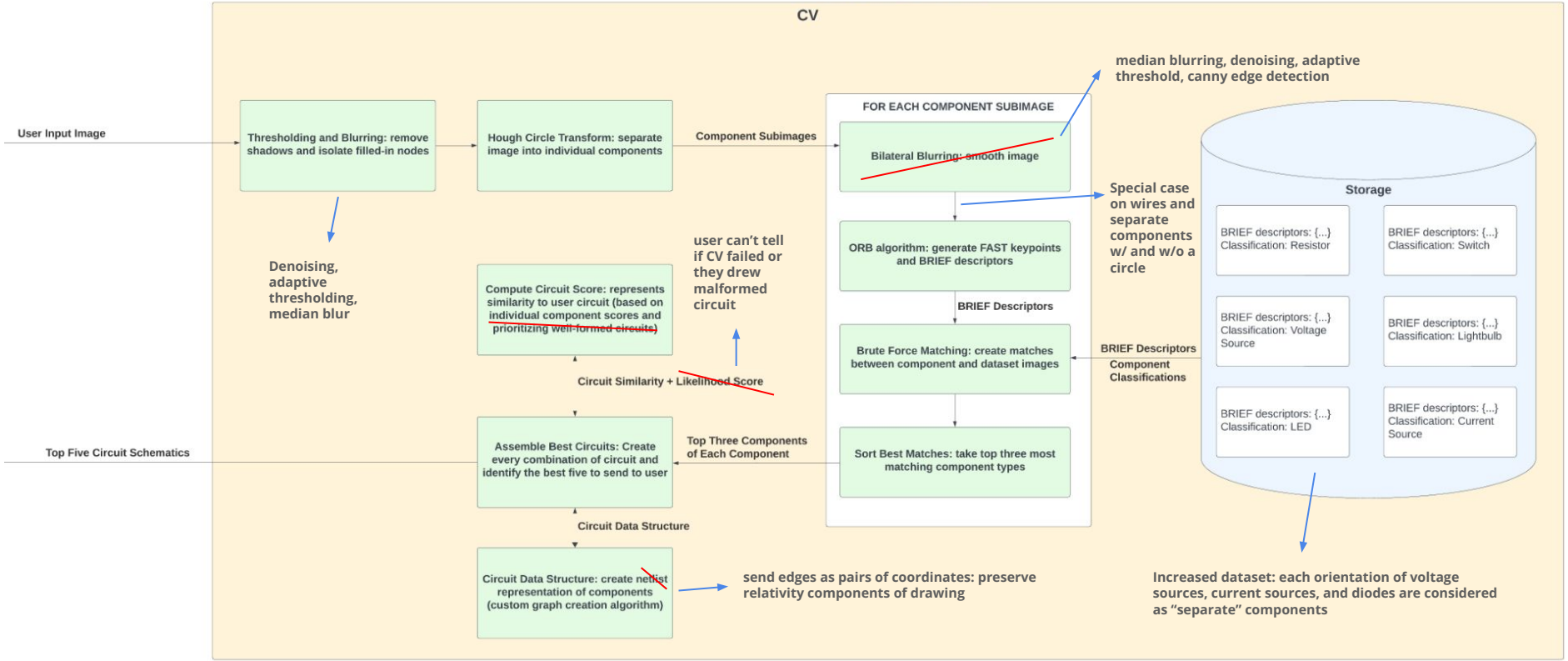



```
Node 1 Voltage: 5.000000 V
Node 2 Voltage: -111.538462 V
Component V1 Current: -388.461538 mA
Component R1 Current: 388.461538 mA
Component I1 Current: 500.000000 mA
Component R2 Current: -111.538462 mA
```

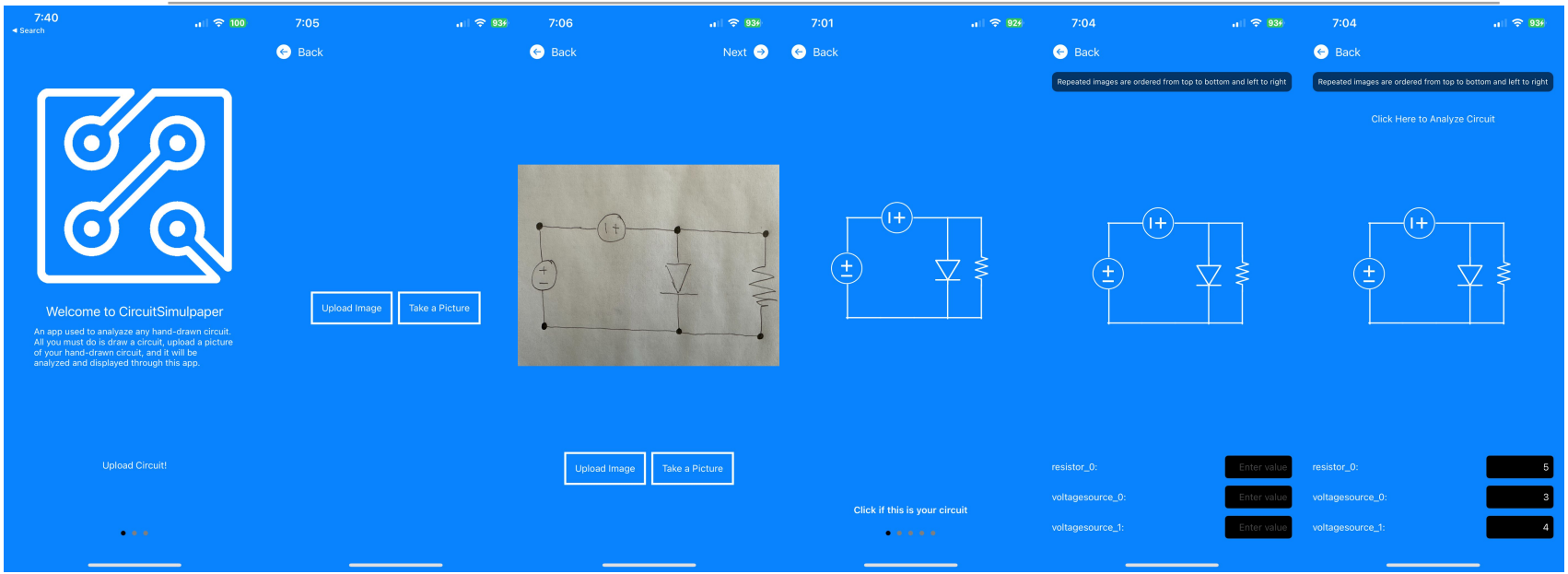
Solution - Mobile Application and Integration

- Bridging of subsystems
- Headers and wrappers
- C++ and Swift to Objective-C++
    - Parsing and casting
- Stored as compatible data type

```
@implementation CVWrapper

+ (NSMutableArray *)run_cv:(NSString *)file_path data_path:(NSString *)data_path{
    std::string fp = std::string([file_path UTF8String]);
    std::string dp = std::string([data_path UTF8String]);
    NSMutableArray *convertedCircuits = [NSMutableArray array];
    CircuitClassifier classifier = CircuitClassifier();
    std::vector<std::vector<Component>> bestCircuits = classifier.getCircuits(fp, dp);
    for (const auto &circuit : bestCircuits) {
            NSMutableArray *convertedComponents = [NSMutableArray array];

            for (const auto &component : circuit) {
                CVWrapper *objcComponent = [[CVWrapper alloc] init];
                objcComponent.firstNode = @[@(component.firstNode.first),
                    @(component.firstNode.second)];
                objcComponent.secondNode = @[@(component.secondNode.first),
                    @(component.secondNode.second)];
                objcComponent.type = [NSString stringWithCString:component.type.c_str()
                    encoding:NSUTF8StringEncoding];

                [convertedComponents addObject:objcComponent];
            }

            [convertedCircuits addObject:convertedComponents];
        }
    return convertedCircuits;
}

@end
```


CV
User Input Image
Thresholding and Blurring: remove shadows and isolate filled-in nodes
Denoising, adaptive thresholding, median blur
Hough Circle Transform: separate image into individual components
Component Subimages
FOR EACH COMPONENT SUBIMAGE
Bilateral Blurring: smooth image
median blurring, denoising, adaptive threshold, canny edge detection
ORB algorithm: generate FAST keypoints and BRIEF descriptors
Special case on wires and separate components w/ and w/o a circle
BRIEF Descriptors
Brute Force Matching: create matches between component and dataset images
BRIEF Descriptors
Component Classifications
Sort Best Matches: take top three most matching component types
Top Three Components of Each Component
Storage
BRIEF descriptors: {...}
Classification: Resistor
BRIEF descriptors: {...}
Classification: Switch
BRIEF descriptors: {...}
Classification: Voltage Source
BRIEF descriptors: {...}
Classification: Lightbulb
BRIEF descriptors: {...}
Classification: LED
BRIEF descriptors: {...}
Classification: Current Source
Increased dataset: each orientation of voltage sources, current sources, and diodes are considered as "separate" components
user can't tell if CV failed or they drew malformed circuit
Compute Circuit Score: represents similarity to user circuit (based on individual component scores and prioritizing well-formed circuits)
Circuit Similarity + Likelihood Score
Top Five Circuit Schematics
Assemble Best Circuits: Create every combination of circuit and identify the best five to send to user
Circuit Data Structure
Circuit Data Structure: create netlist representation of components (custom graph creation algorithm)
send edges as pairs of coordinates: preserve relativity components of drawing

Complete Solution



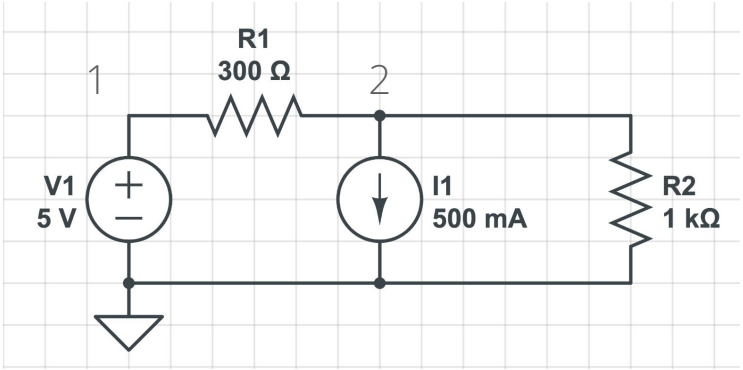
Testing & Verification - Frontend

- Test group of 7 **12-14 year olds**
- Survey UI/UX satisfaction on scales of 1-10
- **Goal: 80%** average rating
- Record feedback regarding
  - Ease of use
  - Usefulness of tips and headers
  - Clarity when displaying values or images

Testing & Verification - Circuit Simulator

- **Goal: 100%** accuracy
- **Result: 100%** accuracy
- Randomly generated netlists with at least one voltage source
  - Tested simulator against existing simulator tools

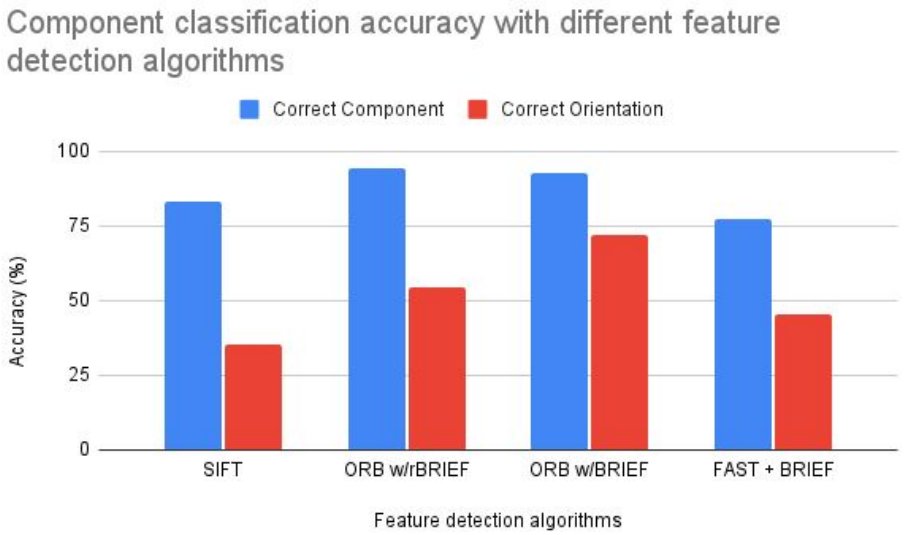
V(V1.nA)	5.000 V
V(I1.nA)	-111.5 V
I(V1.nA)	-388.5 mA
I(R1.nA)	388.5 mA
I(I1.nA)	500.0 mA
I(R2.nA)	-111.5 mA




```
Node 1 Voltage: 5.000000 V
Node 2 Voltage: -111.538462 V
Component V1 Current: -388.461538 mA
Component R1 Current: 388.461538 mA
Component I1 Current: 500.000000 mA
Component R2 Current: -111.538462 mA
```

Testing & Verification - Component Classification

- Testing set of 91 component images
- **Goal: 90%** accuracy
- **Result: 83.3%** accuracy
- Classifying orientation
  - rBRIEF vs BRIEF
- Increase in dataset size -> increase in accuracy

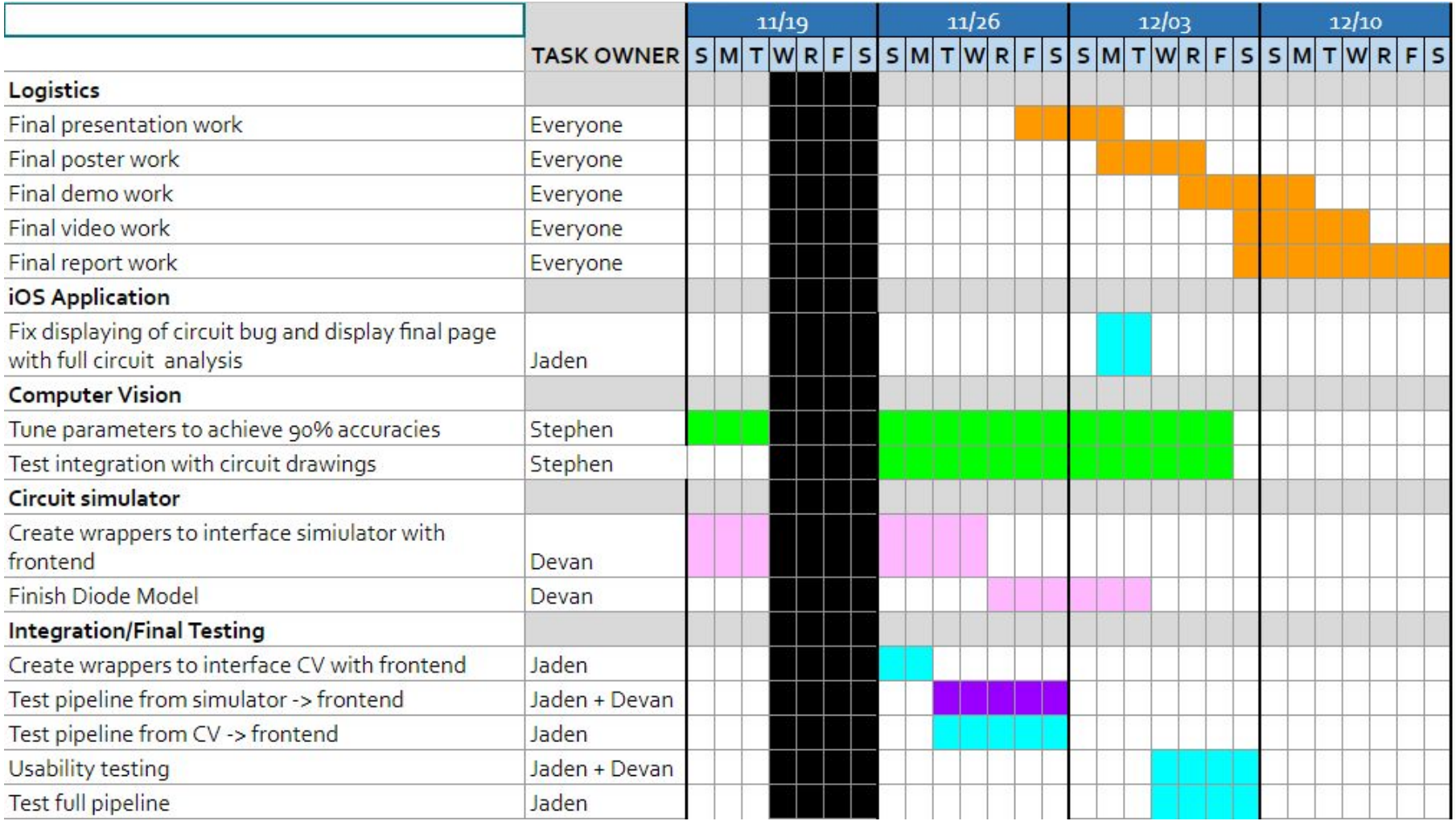


Testing & Verification - Circuit Classification

- Current test set: 27 circuit images
- **Goal: 90%** accuracy
- (Current) **Result: 85.2%** accuracy
- Consideration: neural network
  - Increase accuracy (~95% based on research)
  - Unsuitable for use case
    - Significant storage needed or internet access ($)

Overall Specifications

Requirement	Specification	Current
Application size	<= 100 MB	21.3 MB
Component classification accuracy	90%	83.3%
Circuit classification accuracy	90%	85.5%
Simulation accuracy	100%	100%


11/19
11/26
12/03
12/10
S M T W R F S S M T W R F S S M T W R F S S M T W R F S
TASK OWNER
Logistics
Final presentation work
Everyone
Final poster work
Everyone
Final demo work
Everyone
Final video work
Everyone
Final report work
Everyone
iOS Application
Fix displaying of circuit bug and display final page with full circuit analysis
Jaden
Computer Vision
Tune parameters to achieve 90% accuracies
Stephen
Test integration with circuit drawings
Stephen
Circuit simulator
Create wrappers to interface simiulator with frontend
Devan
Finish Diode Model
Devan
Integration/Final Testing
Create wrappers to interface CV with frontend
Jaden
Test pipeline from simulator -> frontend
Jaden + Devan
Test pipeline from CV -> frontend
Jaden
Usability testing
Jaden + Devan
Test full pipeline
Jaden