**Project One: Prismatica**  
Matthew Dunfee  
Southern New Hampshire University  
CS 360: Mobile Architecture and Programming  
Professor Tyler Parsons  
July 18, 2025

1. Articulate the**goals** of the project.
   1. Describe the application you have chosen, its major components, and what functionality will be necessary based on the scenario.
      * + To bridge this project into a real demo piece, I am staging this project to be consumed by or consume another project of mine. I have an IoT weather station that I can gather data from via Home Assistant. This Android application will serve as a hub for monitoring IoT devices. For the scope of the assignment, I will mock components so they can be staged, and I will also have to bend the roles a little. I’m inventorying data, not material.
        + The design UI components include Login, Signup, Inventory (Read), Inventory Menu (CRUD), Inventory Log (CRUD), and databases (User, Inventory, Inventory Log, maybe more). This will allow me to meet the requirements of the rubric and stage the application to be injected with IoT data later.
        + (describe the functionality)
2. Describe the **users** of the application and the assumptions being made regarding their needs and preferences.
   1. As you think about potential users, consider the following questions to guide your response:
      1. How many different types of users can you identify?
         * Network Owners
         * Network Guest
         * Network managers
      2. What different goals might individual users have?
         * Creating New Devices
         * Update existing devices
         * Remove Unwanted Devices
         * Observe devices
         * Update the existing device's history
         * Removing Unwanted Devices History
         * Observe devices history
      3. What will users need to achieve their goals in this app?
         * Login and password
         * Role assignment (Least Privilege)
         * A network and devices
3. Discuss the screens and features that will be necessary to produce a user-centered **UI design** for the app.
   1. Include a high-level description of every necessary screen and feature that will be in your app.
      * + These are all screenshots of my wireframe for the app. They list the functional objective and have a design for what the UI will look like.

A screenshot of a login screen

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Explain how a user might move from one screen to another.
   1. You may choose to support your ideas with a simple diagram or illustrations to better represent the different components.

A screenshot of a computer

AI-generated content may be incorrect.

1. Justify your decisions by referencing the Android Design and Quality Guidelines linked in the Supporting Materials section.
   * + - After signing in on the Login screen, the user lands on the Inventory list where the primary action “+” FAB creates a new item and tapping any card opens the same CRUD form in edit mode; navigation back to the list always uses Android’s built‑in Up/back hierarchy (no extra on‑screen back buttons), Login and Signup reuse the same form layout to minimize cognitive load, and all buttons, FAB, cards, elevations, and touch targets adhere to Material Design and Quality guidelines—so there’s no need for a separate logout button to satisfy the rubric.
2. Discuss how the functional app requirements will be represented in the **code design** and connected to the UI.
   1. Explain the calls that show the flow of data between code and screens.
      * The page transfers are triggered as follows:

***- page***

***- How to get here***

***- How to leave here***

- Login

- Landing page

- (logout not included in scope)

- Sign in

- Click signup on login

- Submit clicked and form accepted

- Inventory

- Submit clicked on login, and the auth works

- Click the back button

- Inventory Menu

- Click on the ellipse on the inventory/device card or the plug card

- Click the back button

- Inventory Log

- Click on the ellipse on the inventory/device card itself

- Click the back button

* 1. List the major UI components on each screen.
     + This is what I said in the other assignment; the course doesn't give a roadmap or flow chart, or anything to examine what common component hierarchies look like in this ecosystem. For now, I am planning on using stuff like a “groupView”, “View”, “textView”, “painText”, and hopefully some ‘cards’ for styling.
     + Regarding the parent question, the major components are seen in the screen shots and are: Login, Signup, Inventory, Inventory Menu, and Inventory Log.
  2. Determine what data each component will either display or accept as input and where the data might come from.
     + This is further extended to what I said; the lack of examination of component hierarchy leaves me with a lack of understanding of how components fit together functionally. How does one define a flow, a chain, sequence of functions, without understanding how the different pieces fit together to form the overall structure?
     + I can say I will use a “groupView” > “View” > “textView”, “button”, etc., and I'll inject logic via a controller to either the View or the Model, but realistically, I can't instantiate each component. I haven’t learned enough about the implementation of MVC through hands-on experience in this class. This week’s assignment was to put together a non-functional View.
     + My only other assumption is that the question is referring to the function definition of like “F(string, int, float){ return: string}”. If that is the case… To try and define a working road map would be extremely risky, and that goes against "just barely good enough". It would be bad practice when we don’t even know what to full stack consists of.
     + But for posterity,
       1. Login (string username, string password)
       2. Signup (string username, string password, string password)
       3. Inventory (object userAuth)
       4. Inventory menu () <- The prime example of why you can’t predefine functionality without knowledge of the software architecture. It is unknown if there will be the use of arrays in the database, which would allow for all inventory items to have unique field parameters.
       5. Inventory Log () <- same issue at the Inventory Menu, the data isn’t necessarily uniform.