**CrisprMaestro**

The software plan consists of 1.) architecture and 2.) design. Architecture comprises technical details of the apps, the network, mobile devices, the cloud, etc. Design comprises the software solution's capabilities, such as workflow steps and each step's details. While the software architects and engineers specify the architecture, users and stakeholders and technical people all contribute to the design.

**Architectural Assumptions**

The network is reliable and available. The network has low latency and high bandwidth. The hospital firewalls grant the CrisprMaestro services the necessary access to the hospital’s 1.) data, 2.) web services, and 3.) medical devices. The network supports deployment of, and access to, IoT services.

**Microservice Architecture**

The microservice architecture comprises 1.) a user app, 2.) an admin app, 3.) cloud services, 4.) data stores, and 5.) IoT services. The user app runs on a tablet (android, surface, iOS) or a browser. The admin app is a private website accessible to browsers in the hospital network. The apps communicate with Azure cloud services. These Azure cloud services provide security, access to hospital systems and data stores, and communication with IoT services. These IoT services deploy to the hospital private network, and talk to medical devices, such as a centrifuge or Crispr machine.

**WireFraming the Design**

A wireframe is an interactive storyboard of the app. This wireframe lets planners and stakeholders work together to mockup the app's capabilities. Wireframing allows nurses and staff to drive design changes without incurring the cost of modifying live-production code that has already been written and deployed to mobile devices and the cloud. Ideally wireframing is completed before actual software development begins.

**Wireframe the Design with Figma**

A UI/UX designer uses Figma to create a wireframe of the user experience, and the app's (mock) capabilities. The designer runs a Figma demo for the Nurses and staff who look at the forms and provide feedback about what to modify, add, and delete. The designer tweaks the Figma forms to incorporate these requested changes. This process iterates until the stakeholders sign off on the design. Once the stakeholders sign off on the design, the solution’s capabilities and user-interaction are fully specified and actual software development can begin. The first step in software development is to write tests.

**Unit Testing**

Software developers create a set of automated unit tests that verify the expected behavior of the solution's capabilities as captured during wireframing. These unit tests automatically run every time someone changes the solution's code or design. This "future-proofs" the software, as unit tests will fail if modifications alter the expected behavior of the solution. While unit tests validate code, integration tests (see below) verify the app’s capabilities.

**Development**

Figma auto-generates code for the mobile app. UI developers tweak the generated code to interface with the cloud services. Software engineers code the services to implement the expected behavior captured in the unit test suite. As coding increases the resolution of expected behavior, the developers will reflect these enhancements in the unit test suite.

**Agile**

The **Unit Testing** and **Development** sections above describe one release. There will be three releases to complete a minimum viable product. Each release comprises five 2-week sprints and delivers a defined set of capabilities. At the end of each sprint perform 1.) an end-to-end integration test of the sprint's new functionality, 2.) an end-to-end regression test of all other functionality, and 3.) a live demo of the new functionality for the stakeholders.

**Releases**

Wireframe

Login

Workflows

Workflow Manager

R\_0: Architecture

Azure Subscription

Development Environment

CI/CD

AzureAD | SSO/SAML | Datastore

Docker and Git Repositories

Proof of Concept

R\_1: Login

API Gateway

User Login

Administer Users/Groups

Administer User/Group Privileges

R\_2: Workflow Management (website?)

Administer Workflow Tasks/Sequences

Configure Workflow Sequences

Administer Workflow Permissions

Administer User Privileges

Workflow Notifications

R\_3: Default Workflow (TBD)

Tasks

Sequences

Notifications

Security

R\_4: Instrumentation

Monitor System Health

Handle Errors & Bugs

R\_N: Custom Workflow (multiple releases)

Tasks

Sequences

Notifications

Security

**Proof Of Concept**

To prove the architecture and design, create a login page, a landing page, and a login service.

1.) Use Figma to create a login page and a landing page.

2.) Export from Figma to generate React-Native code.

3.) Deploy the React-Native code to an android phone.

4.) Create a login service that authenticates the user.

5.) Deploy the secure login service to the cloud.

6.) Run the phone app to login the user.

**Summary**

Microservices architecture houses the behavior and state of a solution in the cloud. This reduces the need to deploy patches to mobile devices and laptops. Also, this makes the behavior of the app follow the user, not the mobile device or laptop. Thus, users are not limited to one device, but can log in to any station. While it models the user experience on the device, wireframing also informs decisions around software design, and architecture. It is important to bring any unplanned architecture or software design changes that surface during wireframing iterations to the stakeholders’ attention.