CS Capstone Design

Alpha Prototype Demo Grading Sheet (100 pts)

TEAM: PiWatchers

Overview: The purpose of the Alpha Prototype Demo is to clearly demonstrate the extend to which all core user flows envisioned for the product are supported by the current implementation. The flow of the demo is very natural: you simply introduce each of the major usage scenarios, and then follow through each of them, just as an end-user would in using the product. Grading is based on how completely the current product supports all key functional aspects within a coherent, realistic user flow. Interface refinement, clunkiness, and aesthetics should be ignored for now; the focus is simply on functional ability to complete the user flow.

This template is fleshed out by the team, approved by the team mentor, and brought to demo as a grading sheet.

Overview of major product use cases

Based on the Requirements document and subsequent development discussions with your client and mentor, briefly describe each of the key use cases for your product:

UC1: Docker deployment setup for web application.

This details the process that an administrator would go through to set up the backend and the frontend of our web application on their servers.

UC2: Demonstrate backend services.

This details all the different endpoints that are provided through the login authentication service and mongo manager service. This demonstration details the current authentication process and the process for querying and adding data on the mongo manager service.

UC3: Signing up/Signing in for a new account.

This details the process of a user signing up and signing into the service. The demonstration will provide an overview of the authentication process from a user perspective.

UC4: Viewing building/room metrics.

This details the process of a user viewing live data from a selected building and room. The demonstration provides an overview of the building and room selection process as well as the resulting data being displayed.

UC5: Demonstrate analysis and scheduling modules.

This details the IoT device workflow. This demonstration details how the metrics for the database are gathered and sent to the backend with an explanation of what is going on behind the scenes.

User Flows: Detailed walk-through for each use case:

In this section, we outline the demonstrations of each use case that we have prepared, giving a step-by-step outline of the user flow that would be followed by a real user for that use case.

Use case 1: Docker deployment setup for web application.

User Flow: Step by step overview of user interactions with product

1. Download the backend source code from the PiWatchers's GitHub repository.
2. Make updates to the docker-compose.prod.yml file.
3. Create dockerized environment.
4. Create an API user in the MongoDB environment.
5. Run the command for pulling the frontend image on our PiWatcher docker hub registry to setup the frontend portion of the web application.

Evaluation and Comments:

- ✓ Convincingly demo'd each of listed challenges?
- ✓ Other evaluative comments:

Use Case 2: Demonstrate backend services.

User Flow: Step by step overview of user interactions with product

1. Go over the authentication & data endpoints.
2. Demonstrate signing up for a new account through Postman.
3. Demonstrate signing into the new account through Postman.
4. Demonstrate failed attempt at making a new account when it already exists.
5. Demonstrate failed attempt at logging in when the user account information is incorrect.
6. Demonstrate adding an entry into the mongo manager service.
7. Demonstrate showing all the buildings that have entries in the database.
8. Demonstrate showing all the entries for a particular building.

Evaluation and Comments:

- ✓ Convincingly demo'd each of listed challenges?
- ✓ Other evaluative comments:

Use Case 3: Signing up/Signing in for a new account.

User Flow: Step by step overview of user interactions with product

1. Go over the layout of the authentication screen.
2. Demonstrate signing up for a new account.
3. Demonstrate a failed sign in with incorrect credentials.
4. Demonstrate a successful sign in.
5. After signing in, correctly push to the dashboard.

Evaluation and Comments:

- ✓ Convincingly demo'd each of listed challenges?
- ✓ Other evaluative comments:

Use Case 4: Viewing building/room metrics.

User Flow: Step by step overview of user interactions with product

1. Go over the layout of the dashboard.
2. Demonstrate building selection and rooms being displayed.
3. Demonstrate room selection and the selected room data being displayed.
4. Demonstrate live data updates.

Evaluation and Comments:

- ✓ Convincingly demo'd each of listed challenges?
- ✓ Other evaluative comments:

Use Case 5: Demonstrate analysis and scheduling modules.

User Flow: Step by step overview of user interactions with product

1. Show and describe shell script.
2. Begin scheduler application.
3. Talk through how the system is working as endpoints are being hit.
4. Demonstrate counts changing with volunteer entering the lab.
5. Show database through the API.
6. Show it running on current Raspberry Pi in the IoT Lab.

Evaluation and Comments:

- ✓ Convincingly demo'd each of listed challenges?
- ✓ Other evaluative comments:

Known short-comings: Functionality still deficient/missing:

If there were challenges you listed earlier that were *not* covered by a demo, list here. This will hopefully be a short list…but better to be clear about where you are. If you have items here, you could list (if applicable) any pending plans/schedule to get this implemented.

- Role Management System
  - Currently one of the focuses for our milestone is designing and developing a role management system that involves separation of admin/normal user accounts, along with a more complex querying system for our Mongo Manager Service.
- Multiple Charts and Favoriting Charts
  - Another focus of our milestone is incorporating the ability for the user to display multiple charts on the dashboard at the same time as well as the ability to save charts to the dashboard for future reference.