

Team Project

Stakeholder Needs and Grading

Version 7.20260116

Motivation:

The well-known Software-as-a-Service (SaaS) model has now expanded to several online services, including Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Hardware-as-a-Service (HaaS). With the cloud now supporting these and many other services, the term Everything as a Service (XaaS) was coined to refer to the extensive variety of emerging services and applications that people can access on demand over the Internet as opposed to being housed on-premises. According to a new report by Research Insights titled "Global Everything-as-a-service (XaaS) Market Research Report 2019-2026", the XaaS market is expected to grow to more than \$345 billion (about \$1,100 per person in the US) over the next few years. Many industries are now adopting the on-demand approach of acquiring services through cloud computing because it offers agility and flexibility, allowing companies to acquire technology quickly and with fewer up-front costs than they would with a purchase and license agreement.

Team Project:

In this project, we will bring together the tools and techniques (marked in *italics* in this document) that we are learning in the class to implement a Proof of Concept (PoC) for a web application (henceforth, referred to as the app) for a functioning HaaS system. This PoC app is inspired by the University of Utah POWDER program (<https://powderwireless.net/>). The overall nature of the app is predefined, but your team can be creative in adding features that exceed the *stakeholder needs*.

A simplified software architecture of the app is shown in Figure 1 below

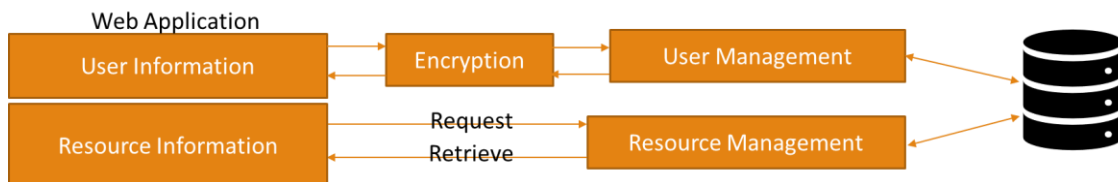


Figure 1: Simplified Software Architecture of a HaaS app

This project will be done with an assigned project team, and you will submit just one deliverable for the entire team during each phase. Each team should contain between 5 and 6 students. All the team members must be in the same section. This will help with coordination as you will have to identify 2 one-hour blocks when you can all meet weekly

HaaS System Project Description

Create a PoC for an app for a functioning HaaS system to enable users to achieve the following *Stakeholder Needs* (SN).

SN1: Create and maintain secure user accounts and projects on the system

SN2: View the status of all hardware resources in the system

SN3: Request available hardware resources and datasets from published sources

SN4: Once approved, checkout and manage these resources

SN5: Check-in the resources and get status of all hardware resources in the system

SN6: Deliver PoC within *schedule constraints*, with support for scalability

Based on these stakeholder needs, your team will define system requirements. The table below lists some of the system requirements to help you get started. You can use these requirements as a starting point and further refine them and add more. As discussed in the class, you should map each requirement to one or more stakeholder needs. Likewise, you should ensure that all stakeholder needs are met.

System Requirements	Stakeholder Needs Met	Related Course Modules
SR1: PoC shall be delivered within budget and schedule constraint, with periodic updates to stakeholders	SN6	SDLC (Software Development Life Cycle), Agile
SR2: PoC App shall have a front-end web application that allows users to enter inputs and views outputs	All	Web UI Development, React.js
SR3: PoC App shall have a mechanism for encrypting user-id and password	SN1	Python OOP (Object Oriented Programming), Python Modules (Cryptography)
SR4: PoC App shall have a mechanism for creating new projects or accessing existing projects	SN1	Python OOP, Python Modules
SR5: PoC App shall have a database for maintaining user login credentials, project codes, project details, resource details	SN2, SN3, SN4, SN5	Python, MongoDB

--	--	--

Minimum Viable Product (MVP)

The features described next are needed to deliver a *Minimum Viable Product* (MVP) for the PoC. Note that you can be creative in expanding the scope of these features.

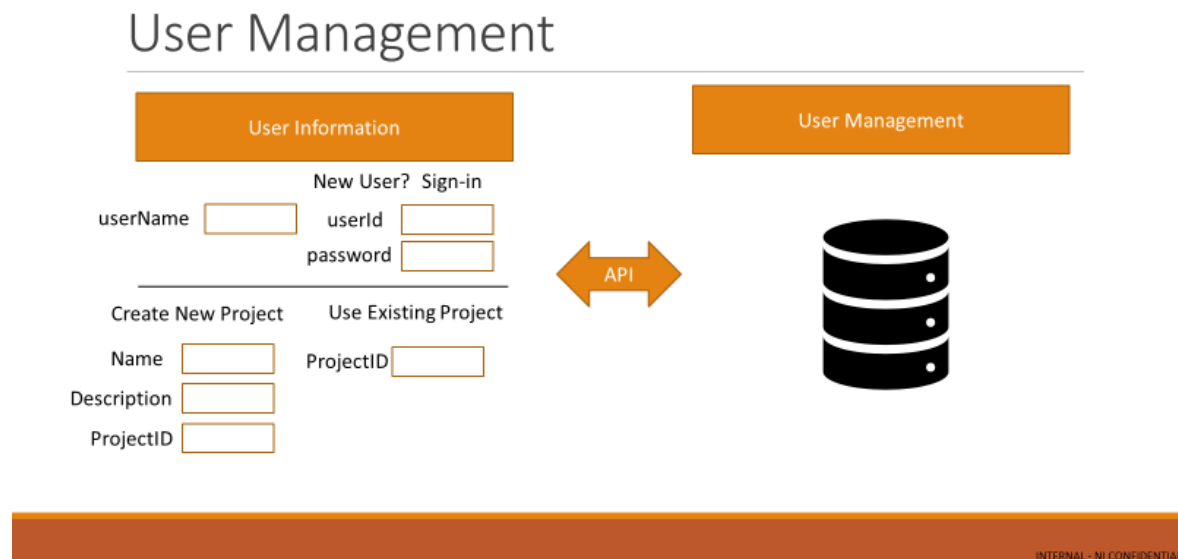


Figure 2: User Management

The user management section, shown in Figure 2, should have the following features:

1. A sign-in area where users can sign in by providing their username, userid and password. If the user clicks on New User, display a pop-up that allows them to enter a new userid and password.
2. An area where users can create new projects, by providing project name, description, and projectID.
3. An area where users can choose to login to existing projects.
4. A database where you can save user information and project information.
5. An API to access information stored in the database.
6. Security features to encrypt the userid and password.

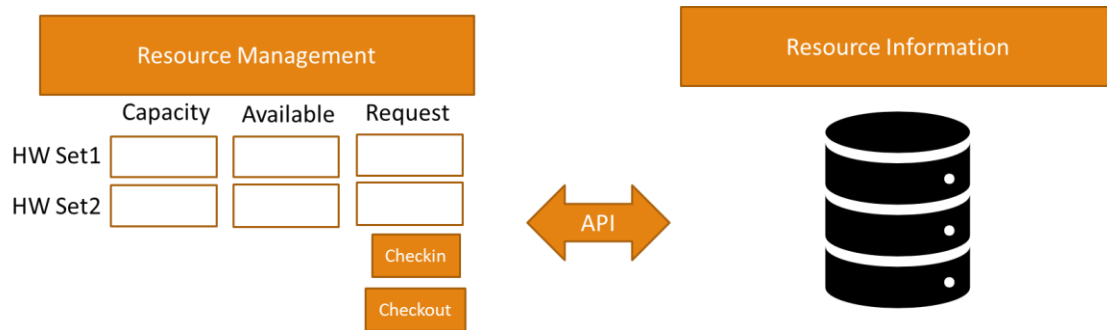


Figure 3: Resource Management

The resource management section, shown in Figure 3, should have the following features:

1. A display area which shows the capacity of HWSet1 and HWSet2.
2. A display area which shows the availability of HWSet1 and HWSet2.
3. A database where the HW information can be stored and can be retrieved from.
4. A display area which shows how many units of HWSet1 and HWSet2 user wants to checkout and later check-in.