Hampton Commerce

Design Specification

Eric Sthurghill, Tori Epps, Joel Magee, Asanji Chofor, James Sanford, Jaelin Jordan, Nicoh Carter

**Table of Contents**

**1.0 Version History** 2

**2.0 Introduction** 3

**2.1 Goals and Objectives** 3

**2.2 Scope of Project** 3

**3.0 Product Design** 5

**3.1 Overview** 5

**3.2 User Interface** 5

**3.3 Expected Input** 6

**3.4 Expected Output** 6

**3.5 Table Description** 6

**4.0 Architectural Design** 7

**4.1 AWS Components** 8

**4.1.1 Simple Storage Service (S3)** 8

**4.1.2 DynamoDB** 8

4.1.3 Amazon Identity and Access Management 8

4.1.3 API Gateway 8

4.1.3 Cognito 8

4.1.4 Lambda 8

**5.0 Restrictions, Limitations, and Constraints** 9

**6.0 Testing** 10

# **1.0 Version History**

****

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Document** |
| 1.0 | 9/19/2019 | Design Specifications Document Version 1.0 |
| 2.0 | 10/10/2019 | Design Specifications Document Version 2.0 |
| 3.0 | 10/31/2019 | Design Specifications Document Version 3.0 |
| 4.0 | 11/18/2019 | Design Specifications Document Version 4.0 |

# 

# **2.0 Introduction**



With new innovative technologies arising in our post-modern era it has made life easier for individuals’ in many aspects. Technologies not only help make people's everyday life easier, but also help companies also when it comes to advertising. Craigslist is an American classified advertisements website with sections such as housing, jobs, for sale, services, and community events. Craigslist was first invented in 1995 as an email distribution list to friends detailing local events in the San Francisco Bay area. It became a web based in 1996 and expanded to other classifieds categories. In 2000, Craigslist started expanding all over the U.S. and covering 70 countries.

Hampton Commerce has been tasked with the assignment to continue the development of the equivalent of craigslist for Hampton University. The application will be usable only by Hampton University students, staff and faculty but anyone could view the website. To insure only members of Hampton University have access, the application will require valid Hampton University credentials, which gives users the ability to sell merchandise, buy merchandise or contact the sellers about their products. The web application will have similar features of craigslist that includes information about events going on in the community, products for sale, housing, services such as tutoring, job opportunities regarding Hampton University.

Hampton Commerce through cloud computing that will be completely hosted on Amazon's AWS which proves several different components that the team will be using.

## **2.1 Goals and Objectives**

The goal of our team’s project is to create the equivalent of craigslist for Hampton University. Within this application, we will need to insure only Hampton University Students’ have access by allowing only a valid Hampton login to be required to interact with the site. Anyone should be able to access this site, but in order to post or contact other users you must have a Hampton University login.

The main goal of this product is to learn about cloud computing. Each team must use AWS service and determine which components will be used to implement the team’s craigslist site.

## **2.2 Scope of Project**

Hampton Commerce has been given an outline of tasks that we must complete in order to finish the craigslist web application for Hampton University that has been started by a previous group (Violent Storms):

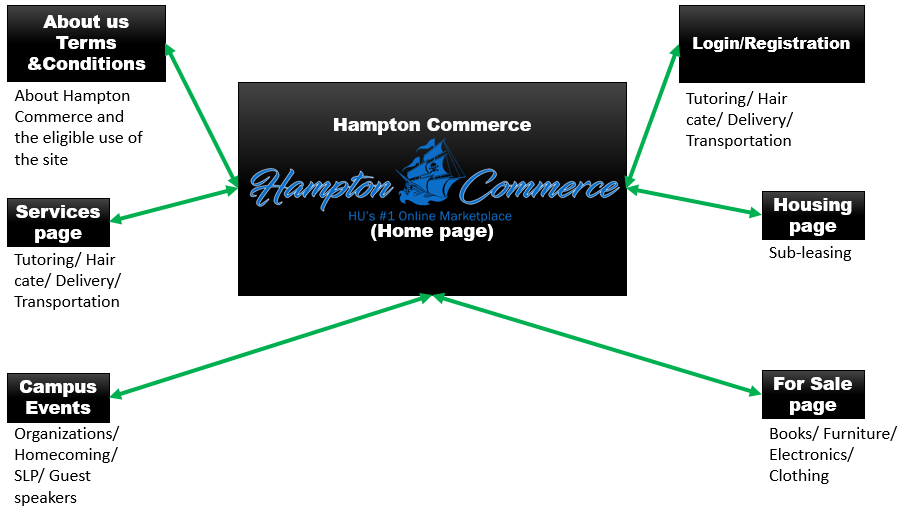
* Create web application through Cloud Computing.
* The application must be hosted in Amazon's AWS.
* Features must be similar to craigslist.
* Valid Hampton University login in order to buy/sell merchandise or contact users.

The scope of this project will be developing a web application specifically for Hampton University that is very similar to craigslist. Hampton Commerce’s website will be completely hosted on Amazon’s AWS through several different components.

# 

# 

# **3.0 Product Design**



## **3.1 Overview**

The product will be designed with the requirements in mind for the development process. The design of the product will consist of two main areas: cloud computing and components within AWS. Cloud computing will allow users and consumers to use applications without installation and access their personal files at any computer with internet access. AWS components will be used for database storage, content delivery, to host our services on a private network and other functionalities to help our business scale grow.

## **3.2 User Interface**

The front-end design of our craigslist website will have a simple and user-friendly interface. The application will have sections categorized in a clear, concise and in chronological order so that users will not get confused on where to find information and posts. Anyone who views our web application will be able to search, buy, or sell, but only valid members of Hampton University will be able to access these exclusive features. The web application will allow cleared users to interact and communicate with other members of Hampton University who then will be able to post information, explore, and most importantly to help one another. In addition, Hampton Commerce will be utilizing Adobe XD to create the front end of the site from the home page, services, and about us page. AWS Cognito will be used for the login/registration page as well as user authentication. The back end will consist of code mainly focusing on HTML and the integration of AWS.

## **3.3 Expected Input**

The expected input for the DynamoDB tables will be the object listings on the web application. These elements would be separated in five (5) separate tables (Services, Housing, For Sale, Campus Events) with these appropriate properties:

* **Object Listings**
  + Post Date
  + Location
  + Description
  + Item/Service/Event/ Housing type

## **3.4 Expected Output**

The expected output would be the listings in order and different posts that will be displayed on each respective page on Hampton Commerce

## **3.5 Table Description**

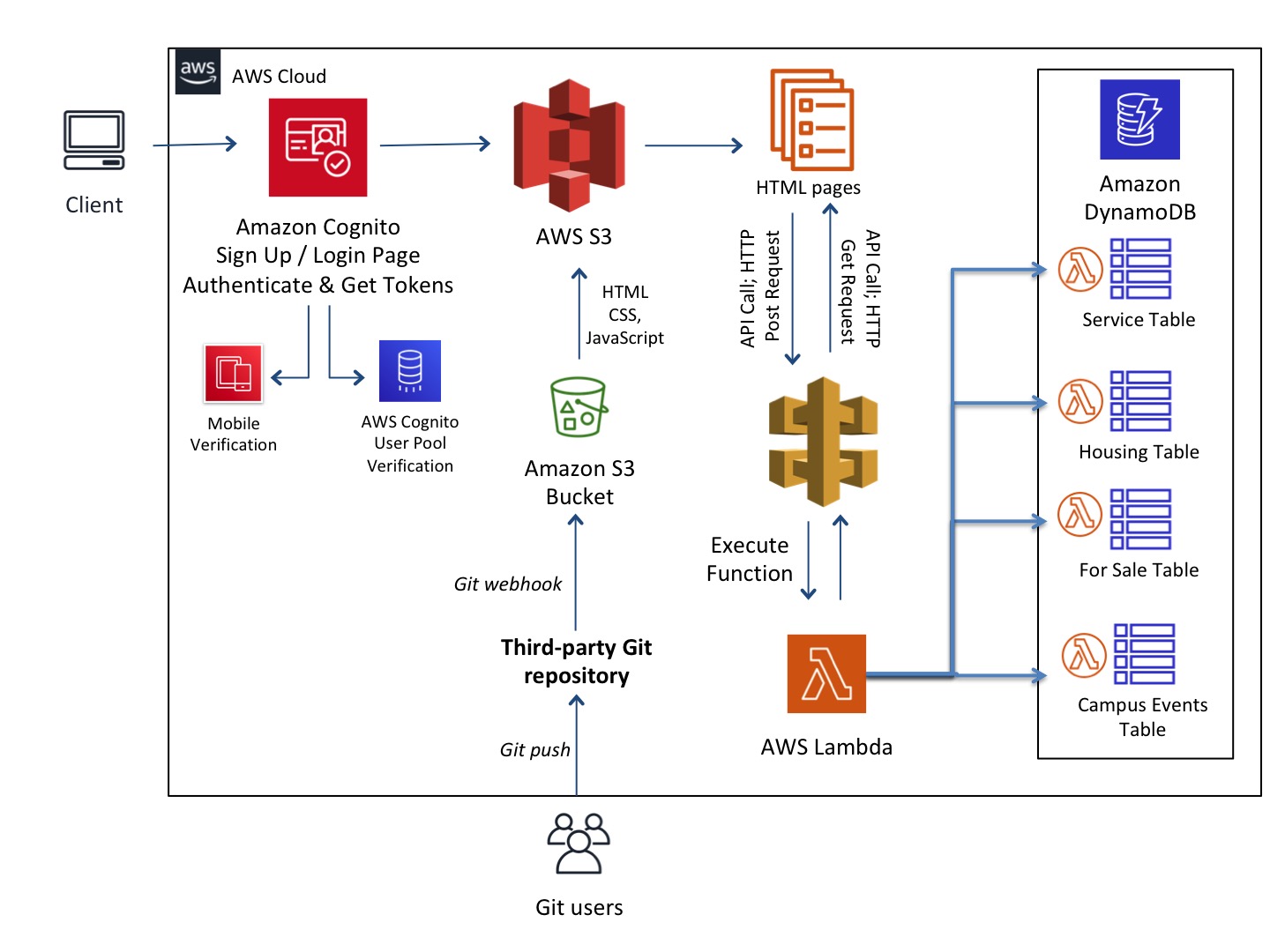
Throughout the next few iterations Hampton Commerce will be enabling a database that will be using AWS DynamoDB. It will migrate the “users” that were created in our application and put them in a table. The web pages will also be in a table as well, where the database table will primarily consist of:

* **Post Date –** Date a specific post was actually and added to Hampton Commerce
* **Location –** When the Item,Service, Event, or Housing type would actually be held
* **Description –** General explanation of the Item,Service, Event, or Housing type
* **Item/Service/Event/ Housing Type –** Each database will have its own specific “type” of listing

# 

# **4.0 Architectural Design**

## 



Hampton Commerce architectural diagram consist of several different components of Amazon AWS. The architectural design starts off with the user utilizing the web application on workstations either on a laptop, desktop. We have decided on a Server less Web Application approach that uses Amazon Cognito for the login page in order to implement two factor authentication via mobile and a token sent to an email verified by our user pool. Amazon S3 will be used to store our Web Pages, HTML and CSS, in an Amazon S3 Bucket and using JavaScript executed in the browser we will send and receive data from a public backend API built using Lambda and API Gateway.

There will be several Lambda functions that correspond to each Dynamo DB Table in order to add, read from and query each database. Source code is stored in GitHub. GitHub is the best option for version control, and most of the team members of Hampton Commerce understand the use of GitHub. Committing and pulling code will be easy, and can be done by using command line, the GitHub website, or the GitHub desktop application.

## 

## **4.1 AWS Components**

### **4.1.1 Simple Storage Service (S3)**

This is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific requirements. S3 uses publicly available web interfaces like SOAP, REST and BitTorrent to provide storage service. This is where we will store our HTML files generated from AdobeXD.

### **4.1.2 DynamoDB**

Amazon DynamoDB is a fully managed proprietary NoSQL database service that supports key-value and document data structures and is offered by Amazon.com as part of the Amazon Web Services portfolio. DynamoDB exposes a similar data model to and derives its name from Dynamo, but has a different underlying implementation.

#### 4.1.3 Amazon Identity and Access Management

Amazon Identity and Access Management, or IAM, allows you to manage access to compute, storage, database and application services in the AWS cloud. IAM uses Access Control Concepts you already be familiar with, such as users, groups and permissions, which get applied to individual API calls. So you can specify permissions to control which users can access specific services, the kinds of actions they can perform, and which resources are available, ranging from virtual machines, database instances, and even the ability to filter database query results.

#### 4.1.3 API Gateway

Amazon API Gateway is an AWS service for creating, publishing, maintaining, monitoring, and securing REST and WebSocket APIs at any scale. API developers can create APIs that access AWS or other web services as well as data stored in the AWS Cloud.

#### 4.1.3 Cognito

Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.

#### 4.1.4 Lambda

An event-driven computing cloud service from Amazon Web Services that allows developers to program functions on a pay-per-use basis without having to provision storage or compute resources to support them. One of the main benefits of AWS Lambda is that it abstracts server management away from the IT professional.

# 

# **5.0 Restrictions, Limitations, and Constraints**

# 

* The Hampton Commerce web application must have a way to validate users that they are members of Hampton University in order to access certain features of the application.
* Establish appropriate databases to store information.

# 

# **6.0 Testing**

# 

The following are the testing methods that will be specified further in the testing document.

* *Unit Testing*
* *Integration Testing*
* *System Testing*