**Systems Design Document**

According to Didacus Odhiambo at Medium.com, there are five major components in a system:

“

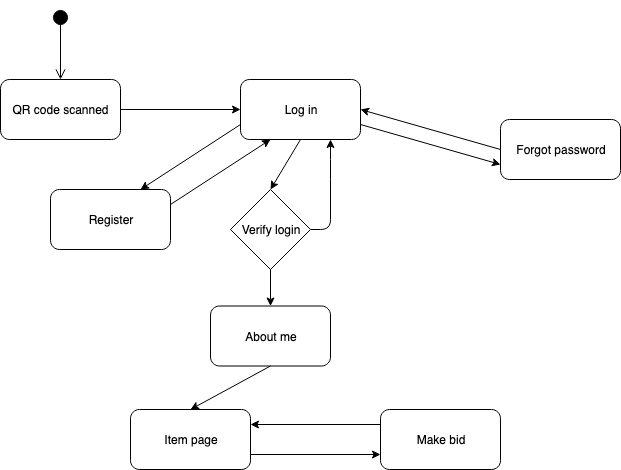
* ***Architecture*** *- This is the conceptual model that defines the structure, behavior and more views of a system. We can use flowcharts to represent and illustrate the architecture.*
* ***Modules -*** *This are components that handle one specific tasks in a system. A combination of the modules make up the system.*
* ***Components -*** *This provides a particular function or group of related functions. They are made up of modules.*
* ***Interfaces -*** *This is the shared boundary across which the components of a the system exchange information and relate.*
* ***Data -*** *This the management of the information and data flow.*

“

[*https://medium.com/the-andela-way/system-design-in-software-development-f360ce6fcbb9*](https://medium.com/the-andela-way/system-design-in-software-development-f360ce6fcbb9)

**Architecture**

Our auction system is built on Django, an ORM (Object Relational Mapper). As part of its platform, development is based on models. These models serve as the building blocks of our system’s auctions, items, bids, user accounts, and webpages. The Django infrastructure allows us to map these relationships together in a coherent way. For example, a user can be tied to an auction they are participating in. The basics of our project’s architecture is illustrated by a simple activity diagram:



As illustrated above, it is necessary for our system’s architecture to interweave components such that a cohesive design strategy can be employed.

**Modules**

Our auction system uses several self-designed templates to build our physical web pages. These templates serve as modules for the entirety of our application. Each of our webpages inherit from a base module that defines the basic formatting that is uniform throughout the auction application. Additionally, we have a template for the navigation bar for the profile, item, and explore pages. This modularity allows us to capitalize on code reuse, as we can modify the templates in one location and see the results everywhere else on the application.

**Components**

There are several components in our project. The first is the database and the models. These are how data is stored in our project. Second, there is the Auctions component. This is responsible for the functionality of most of the website and interaction with the auction. Finally, there is the account component. This is responsible for signing up users and interacting with user profiles.

**Interfaces**

As mentioned above, we use ORM to help the system access the different data. Using the Django provided database, it is possible to save data given to the website, such as when a new user is created or a new auction has been created. It is also possible to access said data on the website to login in to a pre-existing account or check the current status of an item, like which auction it's in, what the current bid is or how much time is left for the auction.

**Data**

Data management is done by Django’s built in database. Much like SQL, Django stores information in the form of tables. It has a separate table for each model instance our system uses. For example, our system has a database table for users.