

Outline Planning and Architectural Design

Team Ramrod (*Cody Kubicek, Lucas Novoa, Tom Delebo, Ben Christianson*)

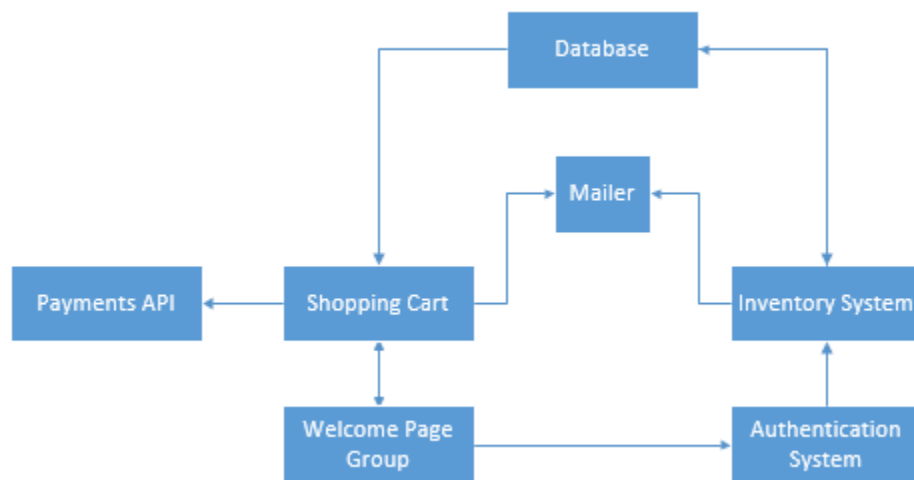
Scope

The intended objectives of project are as follows:

1. Provide a new web presence for Silver and Stones,
2. Provide Silver and Stones with tools to manage their inventory of specialty items, and
3. Provide a means for customers to order from the inventory online.

The system should be easy to use so as to best fit the clientele and employee base (simpler is probably better, when possible). The base system we hope to implement is structured as follows:

Figure 1. Block Diagram of System



The client has requested static web pages to convey information to potential consumers and allow for easy of communication with Silver and Stones. These pages will be backed by an inventory system so that Silver and Stones can manage their inventory, which is mostly comprised of specialty items. A major desire is the ability for employees to update the inventory from their smartphones, which will be accomplished with the use of a mobile-first responsive design. A login system is desired, as is the ability to restrict certain permissions to only certain roles. Eventually, the inventory will include a shopping cart for customers, so that they are able to order items online, which will use the Stripe API to process payment. If there are any messages that need to be passed to users or employees, this will be accomplished via a mailer.

Systems/Tools

We'll be using the Rails framework to implement the system, namely Ruby on Rails. This will be done largely from scratch. Rails is an excellent framework that's supported by the majority of web hosts, but we hope to host it on a cloud server of some sort (e.g. Digital Ocean or Heroku S3) based on what the client is willing to pay for.

As for languages, frameworks, and software development kits, we'll obviously be utilizing Ruby heavily. Of course, this also includes ERB HTML, JavaScript (mostly jQuery), and SASS/CSS. Our work will use Zurb Foundation for a mobile-first responsive framework. We'll also be using Dave Gandy's fantastic Font Awesome icon font, which is fairly ubiquitous amongst web development.

The system will require a database, which can be hosted on the web server. Most web hosts allow for MySQL, but Heroku only supports PostgreSQL. As it stands, we aren't sure what the client is willing to pay for, so we're going into the database design with the intent of using PostgreSQL. However, all of the SQL details are abstracted by Rails' ActiveRecord utility, so we won't have to worry about implementation details until later on.

We will be hitting the Stripe API for payment processing, which is a service that charges credit cards for customers. They charge 2.9% + 30 cents off the top of every "swipe" of the card, but their API is *extremely* easy to use and VeriSign certified. They will handle any issues that may arise with payment.

A private GitHub repo has already been prepared for us. We plan to commit frequently, as Heroku requires git for version control AND deployment. Digital Ocean would allow us to pull from the repo like normal, so it's preferred. In this case, the client could also be kept up to date of any developments immediately (as they could look at the most recent code base).

With Rails, testing is almost required by the framework. We'll be utilizing watir (Web Application Testing In Ruby) and Cucumber to do real-world use case tests on the site. There will also likely be a fair bit of rspec used to test. Any time a major commit is pushed, the testing suite should be run. Cucumber has a nice formatted HTML output format, so we will use

```
cucumber <features> --format html --out <file>
```

to produce testing documentation at each run. These documents will be kept in version control, so we can look back whenever we need to.

Iteration Breakdown

Once each sprint is finished, we will meet with the client and discuss the state of things. We hope to deliver the following at the end of each sprint:

- **First Sprint**
 - Static Pages
 - User Model
 - With Roles Functionality (?)
 - Sessions (Log in)
- **Second Sprint**
 - Inventory
 - Mailers
- **Third Sprint**
 - Shopping Cart
 - Online Orders