Reiman Gardens Butterfly Wing Application

IOWA STATE UNIVERSITY
College of Engineering
EE/CprE/SE 492 - Senior Design

REIMAN GARDENS
IOWA STATE UNIVERSITY

Team Dec1608: Michael Bonpua, Scott Mueller, Carson Noble, Megan Reiman, Nicholas Riesen
Client: Reiman Gardens - Nathan Brockman and Anita Westphal
Advisor: Dr. Diane Rover
Team Site: http://dec1608.sd.ece.iastate.edu

Introduction

This project focuses on creating a new web application for the Reiman Gardens Butterfly Wing. The application needs to be usable both on the Wing's touch-screen interfaces, and on the volunteers' and users' mobile devices. Its purpose is to add a level of interaction to the Wing through images and information portals, and to provide a system for volunteers and staff to manage butterfly data.
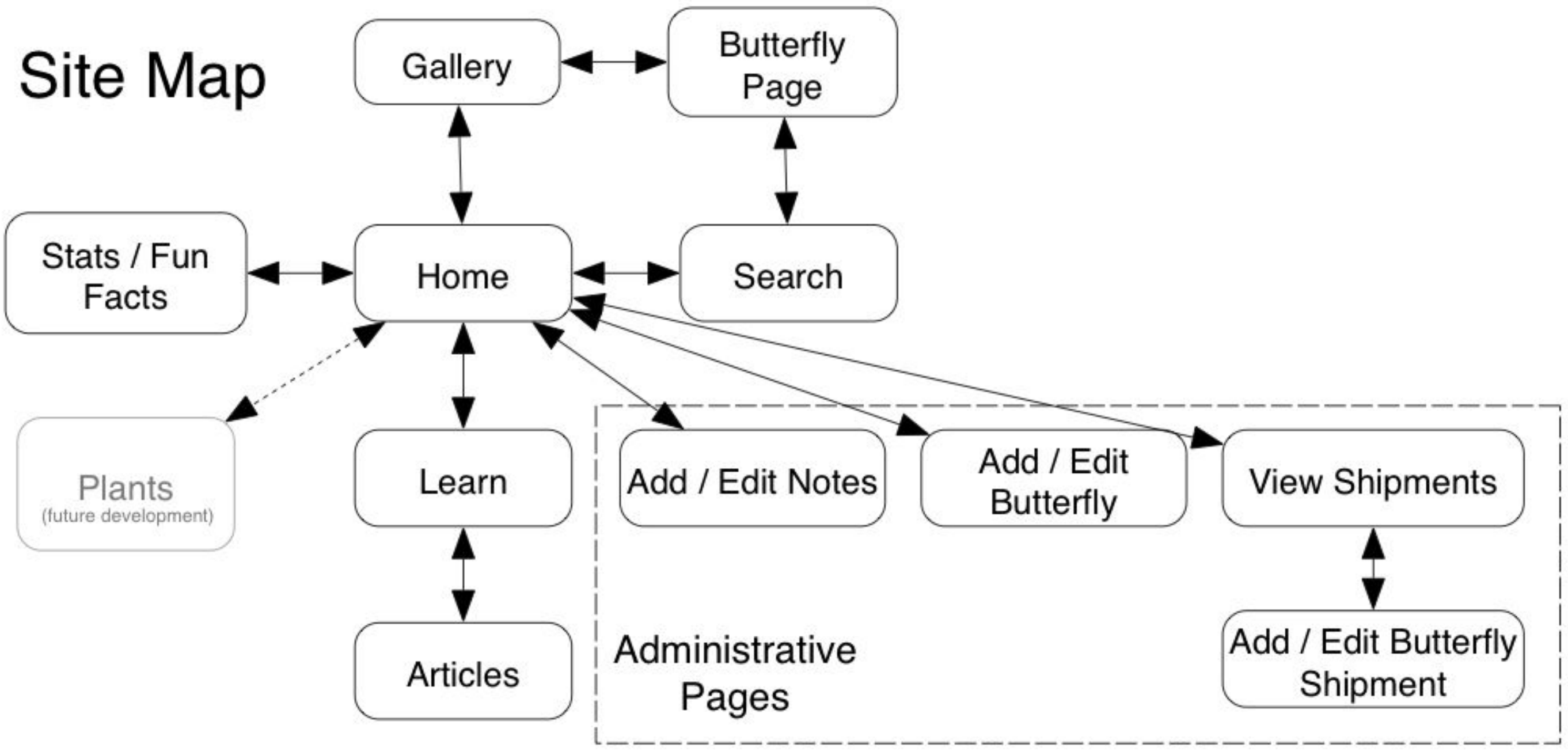
Front End

Goals
- ❖ Responsive to screen size
- ❖ Intuitive navigation
- ❖ Visual appeal

Tools
- ❖ Twitter Bootstrap
- ❖ HTML, CSS
- ❖ JavaScript/jQuery

Site Map

- ❖ The site has intuitive navigation.
- ❖ The administration pages require little to no training to use.

Site Screenshots

- ❖ Content scales based on the device.

- ❖ Administrators go through a simple set of pages to track shipments and releases.

Testing Procedure

Code Reviews: Code is peer-reviewed for efficiency, functionality, and project standards before being merged into the main branch.

User Acceptance: A live demo is hosted on a DigitalOcean server for the client to routinely test and provide feedback.

Functional Requirements

Visitor Side
- ❖ View butterfly and plant photo galleries
- ❖ Search butterflies based on traits
- ❖ View statistics about the wing
- ❖ Read educational pages

Administrative Side
- ❖ Log butterfly shipments and releases
- ❖ Add/edit/remove butterflies
- ❖ Add photos and fun facts

Nonfunctional Requirements

- ❖ Display well on all screen sizes
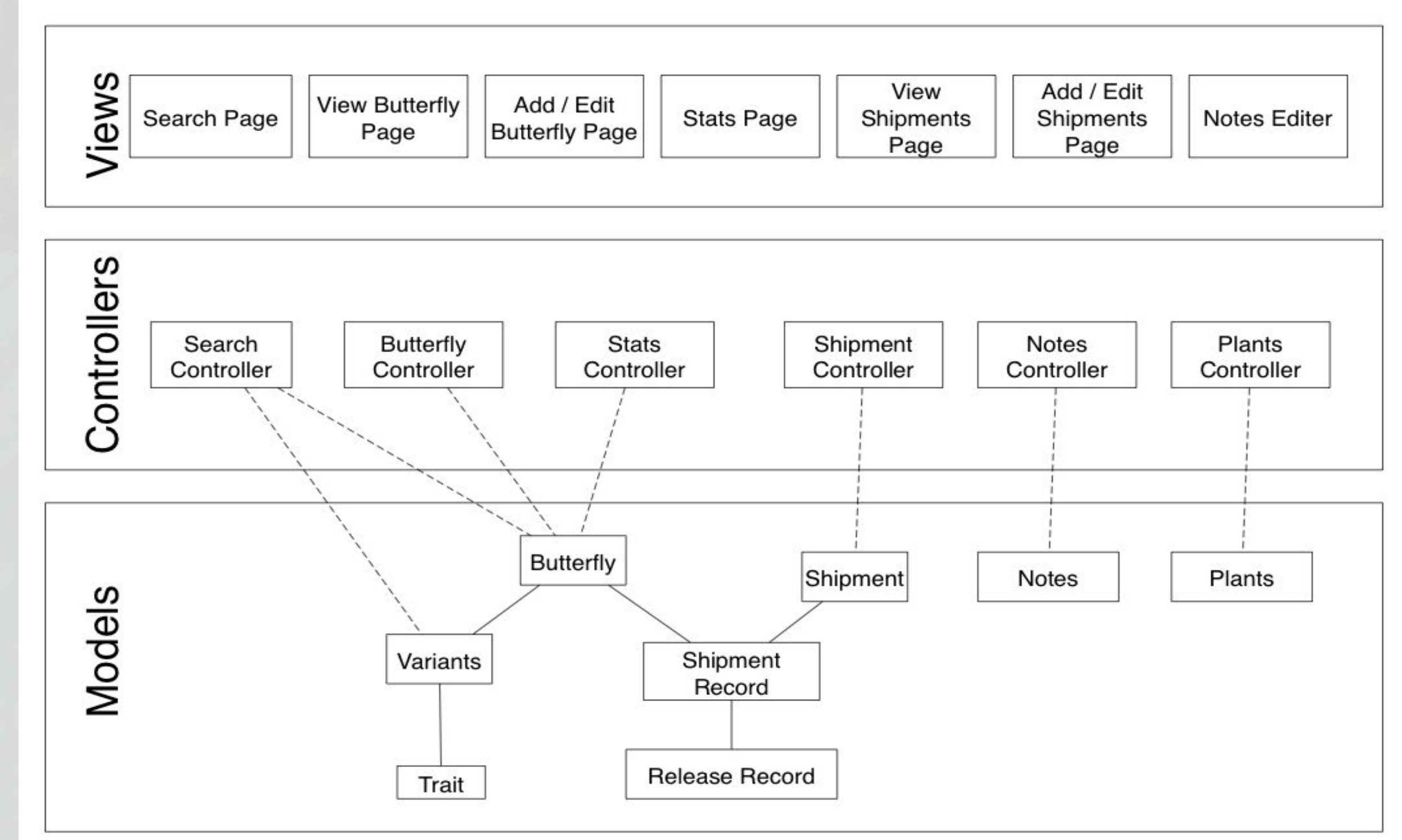- ❖ Intuitive to use for visitors

Operating Environment

- ❖ Butterfly Wing's kiosk (see right)
- ❖ Volunteers' tablets
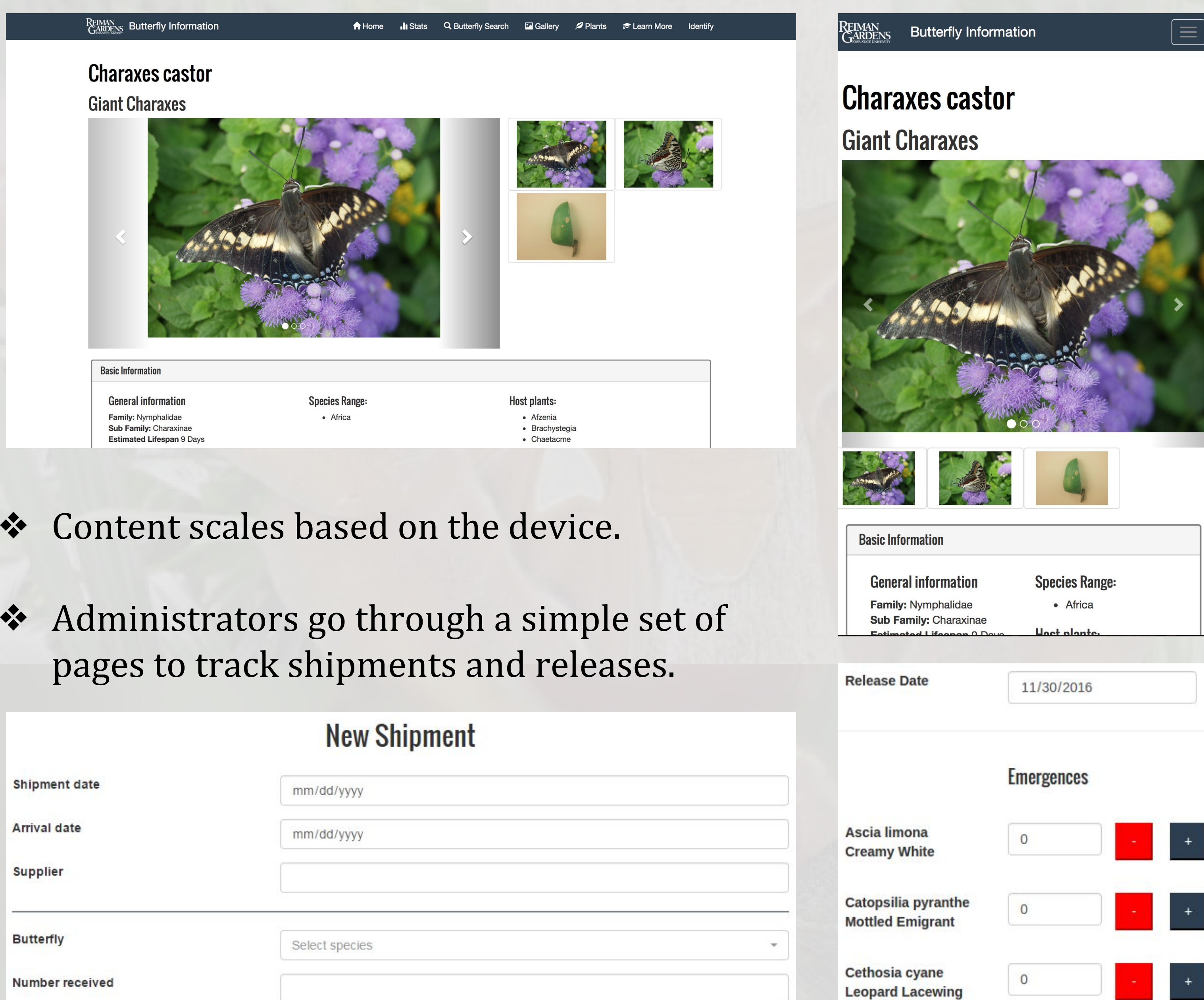- ❖ Visitors' mobile devices

Back End

The application is built using the PHP framework Laravel. It follows a Model-View-Controller architectural pattern, allowing the codebase to be maintainable and structured for continued development.

System Design

Each component consists of a primary model, which supports the functionality to manipulate and display the data associated with it. The model classes are mapped to database tables by Eloquent, Laravel's Object-Relationship Mapping framework.