Quantifier Web Portal

2/26/09
Rev. 1.2

Noah Hilt (Leader/Communicator)
Steven Koechle (Recorder/Communicator)
Joshua Chapman (Facilitator)

Introduction

The International Small Group and Tree Planting Program (TIST) empowers Small Groups of subsistence farmers in India, Kenya, Tanzania and Uganda to combat the devastating effects of deforestation, poverty and drought. Combining sustainable development with carbon sequestration, TIST already supports the reforestation and biodiversity efforts of over 40,000 subsistence farmers in four countries. Carbon credit sales generate participant income and provide project funding to address agricultural, HIV/AIDS, nutritional and fuel challenges. As TIST expands to more groups and more areas, it ensures more trees, more biodiversity, more climate change benefit and more income for more people. As the TISTs effort grows, data delivery becomes more difficult. This project aims to create a new interface for data delivery to the TIST Quantifiers, where smaller sets of more relevant data are delivered to the Quantifier.

Problem Statement

The International Small Group and Tree Planting Program (TIST) helps subsistence farmers in poorer countries plant trees to help sustain their community and reverse deforestation. The current method of information delivery returns the entire data set and does not contain any useful tools for filtering results. The users of TIST's data have a very slow and expensive Internet connection. As the data set grows, it becomes more difficult and more costly for the users to receive data relevant to them.

Solution Statement

The proposed solution is to create an easily accessible application that will allow its users to attain pertinent and relative information in a straightforward manner. For this project, there is a large data set that contains information on the location, count, condition, and other information for the program's trees. It is necessary for this application to allow its users to only obtain information from this data set that is relative to the user. The application must also be easily accessible from anywhere in the world. To solve this problem the application will interface with a web portal that will be available to anyone with an Internet connection. Since this application may be used globally where users may know English as a second language or not at all the solution must include capabilities for translation and/or an interface that does not require written instruction.

Architecture Overview

The solution will be implemented on top of a MVC framework known as Yii. This framework was chosen because its feature set was comparable to many other main stream PHP frameworks, its rating from developers was really high, and some of its benchmarks made it seem much faster than other available solutions. Overall the primary goal is to minimize complexity while meeting the requirements. The structure of Yii creates both a controller and view for every static web page. The controller will handle all logic, while the view will describe how the page looks. In addition model classes will be used to encapsulate database information, such as groves and users. They secondary goal of this project is to provide maintainable code with a platform for continued development. Yii seems to have every available PHP tool that a developer would desire.

The system is broken up into components. Each component is designed to handle a particular focus area, but varies in size depending on the component. Several of these components are simply interfaces to the Yii framework. These components are the GUI Component, Database Component, Authentication/Administrative Control Component, Translation Component, Data Download Component, Advanced Information Component, Report Component and Help/Instruction Component.

GUI Component:

> Generates the HTML that is sent to the user. CSS will be used to ensure easy maintenance and changing of the look and feel. In addition XHTML compliance will be maintained by this component.

Database Component:

> The database access component focuses on encapsulating database access as a set of methods that return results in a sensible fashion.

Authentication/Administration Component:

> Identifies users and uses the administration interface to maintain their settings.

Translation Component:

> Utilizes the internationalization methods of Yii to ensure that all information is displayed in a format the user understands. This includes full text translation as well as formatting of certain information such as dates.

Data Download Component:

> Manages what information is given to the user from the database. In addition it gives access to a way to organize information to be reviewed or printed.

Help/Instruction Component:

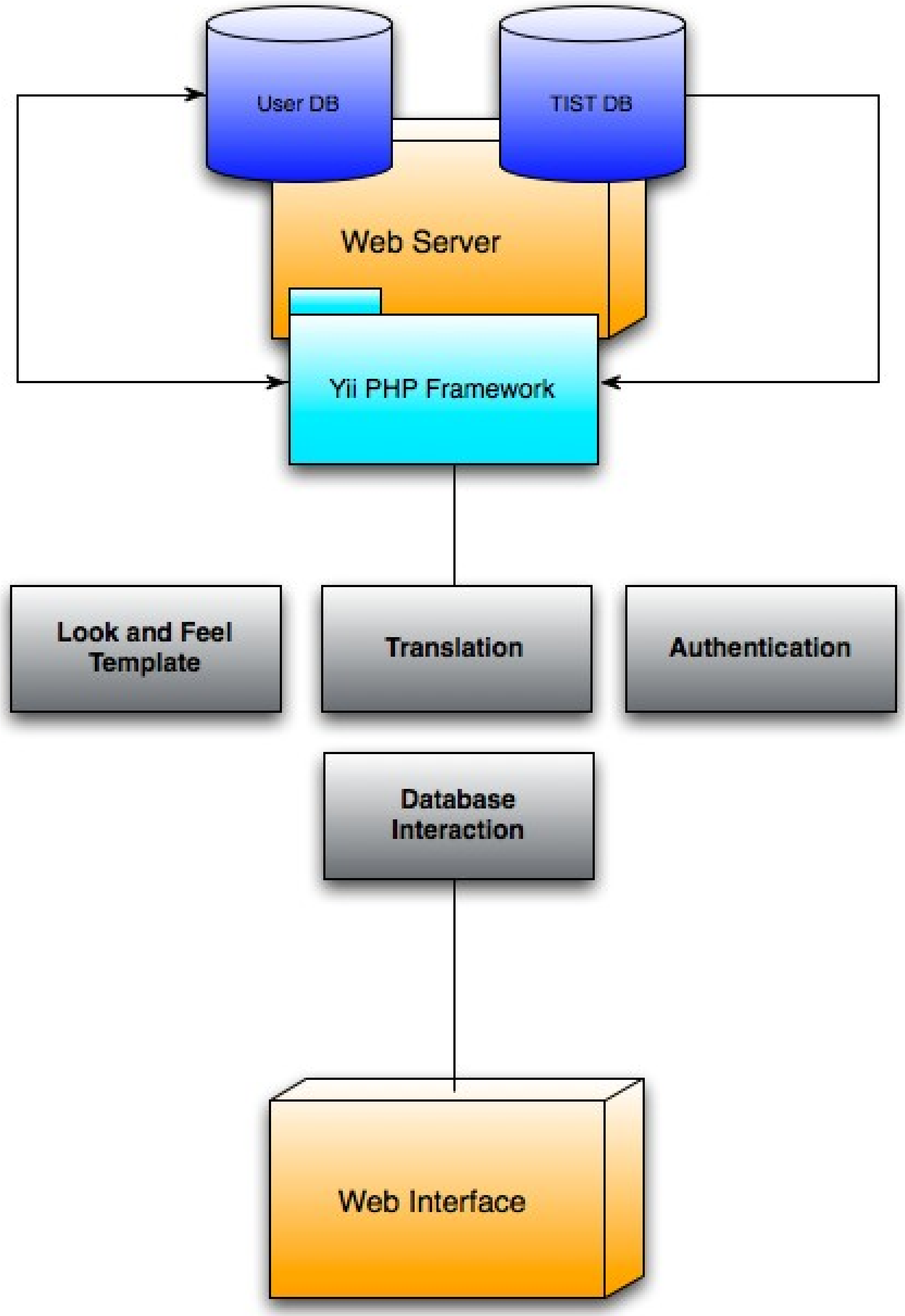
> The pages that gives an explanation of what to do to the users.

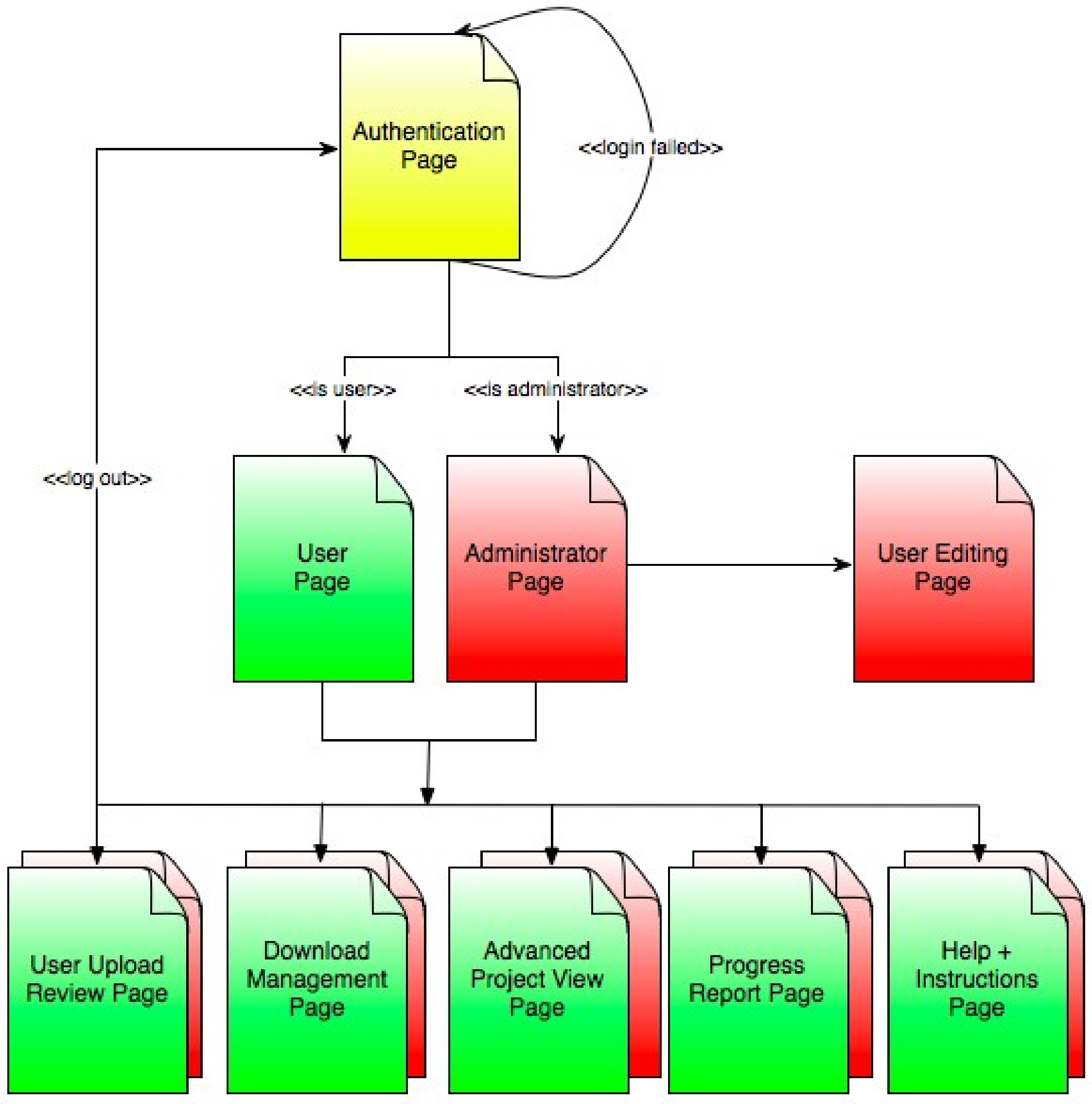
Advanced Project View Component:

A set of alternative views for the project areas that displays more detailed information related to the results.

Report Component:

It filters results based on the current user's uploads. In addition a number of date ranges are available to be filtered based on, with the default being a relatively short time period designed to show the latest set of uploads from the user.

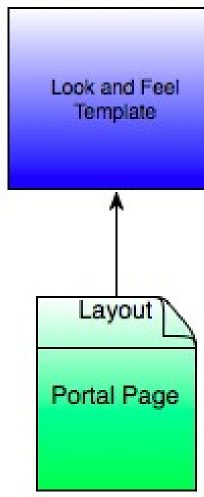

User DB
TIST DB
Web Server
Yii PHP Framework
Look and Feel Template
Translation
Authentication
Database Interaction
Web Interface



Detailed Architecture Description

Look and Feel

Yii provides a templating system for creating a consistent look and feel across an entire site. The layout of the web portal interface will be provided by the Yii framework. This will be defined in a file and accessed by Yii to apply the look and feel to each page in the web portal.



Translation

Yii provides methods for internationalization and translation. Message translation is done by calling a static method - Yii::t(). The method translates the given message from source language to target language.

Authentication/Administrative Component

Yii has built in functionality for authentication and user administration. Extending from this framework, there will be user/administrative authentication as well as an administrative interface to edit users. The authentication will interact with a user database on the web server using the database component.

Database Component

The database component will be focused on creating a simple interface for the system to use to access the database. It will provide an interface that accepts normal objects as parameters, and return normal objects as well. To better simplify things, the component will return either a simple object, an array of objects, or an association (name/value pairs). The backend will be written using the Yii framework's database access objects (DAO). The database component will create a database connection and any other database objects needed for each request, and then format the data before returning it.

Data Download Component

The data download component will be focused on focusing what information is returned on a per report basis. It will perform two primary functions. The first is determining what information the user should be provided in reports about a particular data point. The second is providing organization of data in order to simplify the methodology used by the end user whereby they print a large set of data. This will be implemented as either a flagging method to avoid printing the same information multiple times accidently, or a tab that contains all flagged data points. A possible expansion of this functionality would be to provide a method to print from under the website control, either expanding the flagging method to flag when clicking a print method, or to allow printing all of the flagged data points, depending on what the users find to be the most useful.

Implementation Plan

Implementation will be divided into three phases. Each of these phases will include some portion(s) of the project as well as time for testing and fine-tuning.

Phase 0

This phase will begin with the basic framework for the web portal including the look and feel of the pages, user authentication and database interaction.

Phase 1

After the initial framework of the web portal has been laid out more advanced features will be included as components. This includes upload review, progress report, and advanced project view.

Phase 2

The final phase will be used to implement the help/instruction page as well as complete/touch up any advanced components that have not been finished.

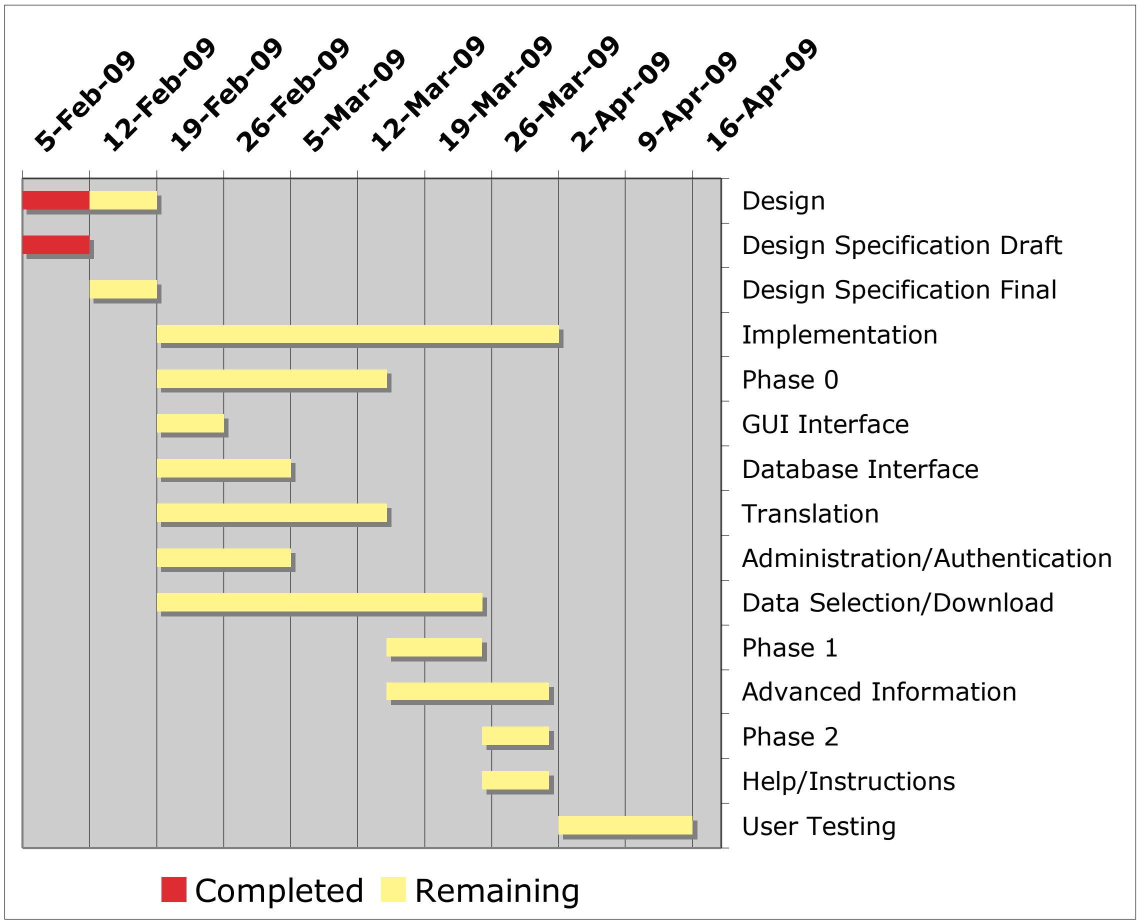


Figure 1. Design Milestone Chart