VINEPILOT

Project Design Specification

v2.0 - The Savvy-gnon Team

Amy Chen - hsuanchen@umail.ucsb.edu
Iris-Eleni Moridis - iriselenimoridis@gmail.com
Richard Waltman - rtwaltman@umail.ucsb.edu
Chien Wang - chienkaiwang0827@gmail.com
Ariel Xin - yuexin@umail.ucsb.edu
# Table of Contents

1. Revision History
2. Introduction
   2.1. Product Overview
   2.2. Glossary of Terms
3. System Architecture
   3.1. High Level Overview
   3.2. Components
      3.2.1. Clients
      3.2.2. Server
      3.2.3. MySQL Database
   3.3. APIs
   3.4. Dependencies
4. Requirements
   4.1. User Stories & Use Cases
   4.2. Prototyping Code & Test Cases
5. System Models
   5.1. General Use Flow Diagram
   5.2. Data Flow Diagram
   5.3. Sequence Diagrams
6. Appendices
   6.1. Technologies Employed
      6.1.1. Web
      6.1.2. iOS
      6.1.3. Server
1. Revision History

v1.0 January 29, 2015 Initial 10 user stories, introduction, system architecture

v2.0 February 26, 2015 Additional user stories, rewritten introduction, system architecture diagram, additional definitions, component & API explanations, diagrams, and employed technologies
2. Introduction

2.1 Product Overview

*VinePilot* is an application that will allow farmers and vineyard owners to interact with aerial NVDI data with as much ease as possible. Users will have the ability to select various NVDI images from a given date, which will be overlayed on top of a map interface provided by Google Maps API. They will also be able to store pins and notes within the application. In the future, users will be able to store information on particular blocks of their vineyards to facilitate the process of screening for ripeness, monitoring the spread of disease and viewing the amount of irrigation. Additionally, they will be able to add notes to any given block in order to further personalize the application and show them the effect of their proactive actions.

2.2 Glossary of Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>VinePilot</em></td>
<td>Name of the application</td>
</tr>
<tr>
<td><em>Server</em></td>
<td>A running instance of an application capable of accepting requests from the client and giving responses accordingly. The server stores all user information and images</td>
</tr>
<tr>
<td><em>Web App</em></td>
<td>Software that runs in a user’s browser - composed of HTML, CSS, Javascript, Bootstrap, and JQuery</td>
</tr>
<tr>
<td><em>iOS App</em></td>
<td>Software that runs in a user’s OSX environment, such as an iPad - composed of Obj-C</td>
</tr>
<tr>
<td><em>Front end</em></td>
<td>An application that the user interacts with directly</td>
</tr>
<tr>
<td><em>Back end</em></td>
<td>An application or program that serves indirect functionality to the user</td>
</tr>
<tr>
<td><em>Obj-C</em></td>
<td>A programming language called Objective C. This is a version of the C language with object oriented design</td>
</tr>
<tr>
<td><em>Javascript</em></td>
<td>An object oriented language commonly used to create dynamic web pages</td>
</tr>
<tr>
<td><em>HTML</em></td>
<td>Hypertext Markup Language - a language that provides formatting and structure for web pages</td>
</tr>
<tr>
<td><em>CSS</em></td>
<td>Cascading Style Sheets - a language to further customize the look and formatting of web pages</td>
</tr>
<tr>
<td><em>NVDI</em></td>
<td>Normalized Difference Vegetation Index - a numerical indicator that uses the visible and near-infrared bands of light to assess different vegetative properties, such as chlorophyll levels</td>
</tr>
</tbody>
</table>
**JSON**  
JavaScript Object Notation - a lightweight data format that uses readable text to transmit data objects of key-value pairs. It is used primarily to transmit data between the client and server.

**HTTP**  
Hypertext Transfer Protocol - a request/response protocol which allows web browsers to fetch information (webpage, pictures, etc) from a web server.

**API**  
Application Program Interface - pieces of software with prewritten functionality that interact with other pieces of software.
3. System Architecture

3.1 High Level Overview

VinePilot is composed of two frontends (web and iOS) and a single backend. The backend, or server, is hosted at UCSB and uses Apache2, PHP, and MySQL to provide the necessary functionality to the front end. This functionality consists of receiving HTTP requests, converting them into the appropriate MySQL queries, and returning the appropriate information via JSON.
3.2 Components

3.2.1 Clients

The web and iOS clients are what the user directly interacts with. These provide the user with the full suite of user interactions, including image selection as well as pin and note additions.
3.2.2 Server

The server’s main purpose is to store data persistently for the users of VinePilot. When the server receives an HTTP request from a user, it will automatically convert the request into the appropriate MySQL request using PHP scripts and return the information extracted from the database via JSON. In addition to storing data persistently and handling HTTP requests, the server will have automatic backup of the database as well as stored images in case of failure.

3.2.3 MySQL Database

The MySQL Database holds all data relevant to VinePilot’s users. This includes NVDI images, pins, notes, and any other user data that will be implemented in the future.
MySQL queries are used to store or retrieve the data that the user currently needs in either the iOS or Web clients. This lives in the server and is automatically backed up in a periodic manner.

The database is organized using three connected tables: Users, Pins and Images. The “Users” table checks authentication upon user login: if the tuple exists in the “Users” table, the user will be logged in. Authentication success will lead the user to the main menu, and ensure that all information retrieved from the database is their own. Additionally, this means their data will be secure.

The “Images” table gives the path to the images, as well as the coordinates to allow for image overlay on Google Maps. The globalNote attribute returns all general notes given by a user, specified by the vineyard attribute. The vineyard attribute is a foreign key referencing the according vineyard in “Users”, and will cascade delete.

The pins table contains all pins currently on the map, and allows the iOS and web application to display the same data. The note attribute is the note that corresponds to a specific pin, which will be displayed once the pin is clicked. The date will allow the applications to only load pins for a specific group of images, ie. images taken on that date. The attributes vineyard, date are a foreign key referencing the according vineyard, date in “Images”, and will cascade delete.
3.3 APIs

The following APIs are used by the web and iOS applications:

- **Bootstrap (Web)** - An HTML, CSS, and JS framework, Bootstrap accelerates the process of styling and coding web pages. Bootstrap provides prewritten CSS for buttons, navigation bars, tables, and much more. Bootstrap also provides easy JS scripting of any intractable object. With bootstrap, web developers can skip much of the back and forth tedious manual styling to save them time. We make use of the following functionality within bootstrap:
  - **NavBar**: `<nav class="navbar navbar-default"></nav>`
  - **Buttons**: `<button type="button" class="btn btn-default navbar-btn">Sign in</button>`
  - **Glyphicons**: `<span class="glyphicon glyphicon-search"></span>`

- **JQuery (Web)** - A JavaScript library that makes dynamic HTML document traversal and manipulation, event handling, animation, and Ajax much simpler to do. We make use of the following functionality within JQuery:
  - Dynamic HTML manipulation - for building the history list after a successful HTTP request

- **Google Maps API (Web & iOS)** - Gives developers methods for embedding Google Maps into their web pages and allows many prewritten functions to dynamically interact with the map. This includes functionality for pins, image overlay, and events. The following functionality from Google Maps API is utilized:
  - **Map Creation**
    - `google.maps.Map(HTML div): map Object`
  - **Map Overlay** - this includes image overlay, pin drops, and user created polygon overlay
    - `google.maps.Marker()`
    - `google.maps.Polygon()`
    - `google.maps.InfoWindow()`

- **CocoaPods (iOS)** - A dependency manager for Objective-C projects. It's goal is to improve the availability of third party open-source libraries that lighten the development load on iOS developers. No CocoaPods libraries have yet been implemented
3.4 Dependencies

As the APIs are not loaded via our server, our application is reliant on the source files included in the HTML being present on whatever server they live on. This applies to Bootstrap, JQuery, and Google Maps.
4. Requirements

4.1 User Stories & Use Cases

*Each user case is implemented twice, on both the Web Application and the iOS Application*

1. As a user, I want to log in so that my data is secure
   a. Time Estimate: 2
   b. Acceptance Test:
      i. **Scenario:** The user opens the application
      ii. **Given** the application is open, **when** the user provides a correct username and password, **then** the application will forward the user.

2. As an application, I want to reject any user with incorrect login information
   a. Time Estimate: 2
   b. Acceptance Test:
      i. **Scenario:** The user enters the wrong login information
      ii. **Given** the user enters the wrong combination of account name and password, **when** the user presses “Log In” button, **then** the application will reject the login and give an error message.

3. As a user, I want to see only my farm’s data
   a. Time Estimate: 4
   b. Acceptance Test:
      i. **Scenario:** The user logs into the application
      ii. **Given** the application is opened for the first time, **when** the user enters the correct login information, **then** the user will be able to see only the specified farm’s information.

4. As a user, I want to have a user-selectable region of the vineyard for different types of grapes
   a. Time Estimate: 10
   b. Acceptance Test:
      i. **Scenario:** The user is on the maps view and wants to create an overlay polygon (and possibly name it)
      ii. **Given** the user is on mapview and wants to create a region, **when** they click on Create Region button **then** the user will be prompted to create their polygon.

5. As a user, I want my password to be hidden while I type so that nobody can see my data but me
   a. Time Estimate: 1
   b. Acceptance Test:
i. **Scenario:** The user is entering their password into the login screen
   ii. **Given** the user is on the login screen, **when** the user starts typing in their password, **then** the password box will only display *'s.

6. As an application, after login I want to go to the menu page
   a. **Time Estimate:** 2
   b. **Acceptance Test:**
      i. **Scenario:** The user successfully logs in with their username and password
      ii. **When** the login is successful, **then** the application will forward the user to the full application.

7. As a user, I want to be able to set notes for my farm
   a. **Time Estimate:** 5
   b. **Acceptance Test:**
      i. **Scenario:** The user is logged in and selects the notes function (accessible via menu)
      ii. **Given** the user is logged in securely, **when** they click the notes menu option, **then** a window dialogue opens for them to write and save notes in.

8. As an application, I need to be able to add the user’s notes to the database
   a. **Time Estimate:** 5
   b. **Acceptance Test:**
      i. **Scenario:** The user is logged in and adds a note
      ii. **Given**, the user has reached the add notes page and has clicked save on their note **then** the note should be added to the database

9. As an iOS application, I need to have a local database to save data added while there is no wifi
   a. **Time Estimate:** 8
   b. **Acceptance Test:**
      i. **Scenario:** The user has no wireless connection and wants to add notes
      ii. **Given**, the user has created the note and saved it **then** save the note locally to a database similar to SQL

10. As an application, I want to pull the list of dates of notes from the database
    a. **Time Estimate:** 8
    b. **Acceptance Test:**
       i. **Scenario:** The user is in image view and wants to view all dates on which notes were taken; click a view notes function to display list of dates to give the user the choice of dates from which to view notes
ii. **Given** the user is in image view and clicks on a *Dates* button then the application should show a list of dates on which images were taken

11. As an application, I need to pull the notes corresponding to a specific date from the database  
   a. Time Estimate: 6  
   b. Acceptance Test:  
      i. **Scenario:** The user logged in and looking for the notes he/she added on a specific (selected) date  
      ii. **Given** the user selects a date and the date exists in the database then the notes corresponding to that date will be returned from the database

12. As an iOS application, I need to be able to save the notes pulled from the database to the local database  
   a. Time Estimate: 6  
   b. Acceptance Test:  
      i. **Scenario:** user is on a *View Notes* page so application has retrieved any notes added to the database since the last time the notes were pulled  
      ii. **Given** that there are notes pulled that do not already exist in the iOS local database (ie. notes were added from the web application) then save the new notes locally

13. As an iOS application, I need to be able to push data from the local database to the SQL database  
   a. Time Estimate: 7  
   b. Acceptance Test:  
      i. **Scenario:** The user has added, edited or deleted a note or pin when they did not have wireless connection  
      ii. **Given** there exist changes in the local database that don’t exist in the SQL database then update the SQL database

14. As an application, I need to pull the list of dates of images for each vineyard  
   a. Time Estimate: 7  
   b. Acceptance Test:  
      i. **Scenario:** The user is logged in and wants to pick a set of images from a list of dates for *history* function to work  
      ii. **Given** the user is in the image or map view and clicks on the “History” button then a list of dates on which images were taken will be pulled for the user to choose from

15. As an application, I need to communicate with the database to get the path to an image of a specific farm  
   a. Time Estimate: 5
b. Acceptance Test:
   i.  *Scenario:* The application opens an image depending on which user is logged in
   ii. *Given,* a user has logged in and their data exists in the database then retrieve the “path” attribute from the correct tuple

16. As a user, I want to be able to select images of my farm throughout the history of the service
   a. Time Estimate: 6
   b. Acceptance Test:
      i.  *Scenario:* The user is logged in and selects the *history* function (accessible via menu)
      ii. *Given* the user is logged in securely, *when* they click the *history* menu option, *then* a list of all available pictures will be given.

17. As an application, I need to load the correct picture when a different picture from *history* is selected by the user
   a. Time Estimate: 5
   b. Acceptance Test:
      i.  *Scenario:* The user is logged in and clicks a *different* picture than what is currently shown
      ii. *Given* the user is in the *history* menu and clicks an option *different* than what they currently see, *then* the application must *get* the correct image from the server and display it.

18. As a user, I want to see my vineyard on a map
   a. Time Estimate: 4
   b. Acceptance Test:
      i.  *Scenario:* The user logs in and goes to *Image View* to view their vineyard on the map
      ii. *Given,* that the user has logged in successfully and they have clicked on some *Image* button *then* the app should show them the map with the location of their vineyard centered.

19. As a user, I want to zoom in and out of my farm
   a. Time Estimate: 5
   b. Acceptance Test:
      i.  *Scenario:* The user is logged in and pinches/scrolls out/in
      ii. *Given* the user is using the application and pinches/scrolls out/in, *then* the map should zoom in and out accordingly

20. As an application, I need the coordinates of the images to overlay them on the map
   a. Time Estimate:
i. if VineRangers have the coordinates: 1 or 2
ii. if not: 8 or 9

b. Acceptance Test:
   i. Scenario: The application’s image view contains the map and needs to also overlay the near-infrared image in the map
   ii. Given, that the image exists and that the SW and NE coordinates exist then the application will overlay the image over the map to allow the user to view the images with geographic understanding of the areas.

21. As an iOS application, I need to save all images locally after they are added as a note or downloaded from the server
   a. Time Estimate: 8
   b. Acceptance Test:
      i. Scenario: User has logged in and wishes to view images that correspond to their
      ii. Given, user images exist in the server then they must also be saved locally to ensure the wireless functionality is maintained for iOS device.

22. As a user, I want to be able to drop pins
   a. Time Estimate: 6
   b. Acceptance Test:
      i. Scenario: The user is logged in and selects the pin function (accessible via menu)
      ii. Given the user is logged in securely, when they click the pin menu option, then a pin marker will replace the mouse and drop on the map when clicked.

23. As a user, I want to be able to add notes to specific pins
   a. Time Estimate: 8
   b. Acceptance Test:
      i. Scenario: A user is looking at an image and wants to write a note relevant at specific coordinates
      ii. Given, the user is on the google maps view and has dropped a pin then add a note for that specific pin

24. As an application, I need to store the coordinates and notes corresponding to a specific pin in the database
   a. Time Estimate: 6
   b. Acceptance Test:
      i. Scenario: The user saves a note to a specific pin that they will later want displayed on both the web application and the iOS application
ii. Given, the user has added a note to a pin and saved it then the note and its coordinates should be stored to the database.

25. As an application, I need to pull the list of pins with their corresponding coordinates and notes the database
   a. Time Estimate: 8
   b. Acceptance Test
      i. Scenario: The user logs into one of the two applications and drops pins on the map, the user expects those same pins to appear on the other application
      ii. Given the user has added pins on one application and has not added them on the other, then the application should get the appropriate information from the database and add those pins.

26. As a user, I want the note I added to a specific pin displayed every time that pin is clicked
   a. Time Estimate: 9
   b. Acceptance Test:
      i. Scenario: A user is logged in securely and wants to see the note at a specific location
      ii. Given, the user has dropped a note to a pin and clicked on it then display the note that was added to that pin.

27. As a user, I want to edit notes I have previously saved
   a. Time Estimate: 8
   b. Acceptance Test:
      i. Scenario: The user is in notes view and finds a note they want to edit
      ii. Given, the user is viewing notes and clicks edit then the edited note will replace the unedited one in the View Notes view.

28. As a user, I want to delete notes I have previously saved
   a. Time Estimate: 6
   b. Acceptance Test:
      i. Scenario: The user is logged in and finds an unnecessary/unwanted note
      ii. Given, the user is viewing notes and clicks delete then the note will no longer appear in the View Notes view.

29. As an application, I need to be able to delete user’s notes in the database
   a. Time Estimate: 5
   b. Acceptance Test:
      i. Scenario: The user is logged in and deletes one of their previous notes
ii. Given, the note exists in the database and the user has clicked delete then the note will be deleted from the database

30. As an application I need to be able to edit user’s notes in the database
   a. Time Estimate: 5
   b. Acceptance Test:
      i. Scenario: The user is logged in and edits a pre-existing note
      ii. Given, the note exists in the database and the user edits it on the application then update the note in the database

4.2 Prototyping & Test Code
iOS: https://github.com/PARanOiA1120/VinePilot_iOS
Web: https://github.com/rtwaltman/VinePilot_Web
5. System Models

5.1 General Use Flow Diagram

This diagram shows the sequence of events that can take place upon user action.

- Introduction page
  - Signup
  - Login
  - Main Menu
    - Select Date
      - Add Global Note
      - Add Pin
      - View Notes
    - View Image
5.2 Data Flow Diagram

This diagram shows how the different functionalities interact with the server and database. The login information will be taken from the database and allow the user to access the server images if the login is successful. This is what is mean by “Account” log in and receive information.

Notes describes how adding notes and viewing notes is dependant on the database; those notes will be added to the database, and then the user can only view them by accessing the database.

The application communicates with the server in order to retrieve the image, which it will then display for the user to see as it will be overlayed on the map.
5.3 Sequence Diagrams

Here are sequence diagrams to describe the events that will take place based on the following actions of the user: selecting date of image to overlay, dropping a pin on an image, adding notes to their current date, and logging out.
6. Appendices

6.1 Technologies Employed

6.1.1 Web

JavaScript  An object oriented language commonly used to create dynamic web pages

jQuery  A JavaScript library that makes dynamic HTML document traversal and manipulation, event handling, animation, and Ajax much simpler to do.

Bootstrap  An HTML, CSS, and JS framework, Bootstrap accelerates the process of styling and coding web pages.

QUnit  A JavaScript unit testing framework that supports server-side and client-side environments

6.1.2 iOS

CocoaPods  A manager for objective-C language that provides standard formats for citing external libraries

6.1.3 Server

Apache2  An open source HTTP Web server

PHP  A server-side scripting language designed for web development

MySQL  A Relational Database Management System (RDBMS) that uses Structured Query Language (SQL) for accessing and managing content in a database