****

**LVS Connect**

**Shelby Smith and Derek Bailey**

**April 24th, 2018**

**CEN 4935 Senior Project - Spring 2018**

**Instructors: Dr. F. Gonzalez & Dr. J. Zalewski**

**Department of Software Engineering**

**Florida Gulf Coast University**

**Ft. Myers, FL 33965**

**Copyright © 2018 by Florida Gulf Coast University**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lee Virtual School Mobile Application** | | | |
| **No.** | **Student’s name** | **Contribution in percentage** | **Verbal description of the contribution** |
| **1** | **Derek Bailey** | **50%** | **GUI Mockups, leads application design.** |
| **2** | **Shelby Smith** | **50%** | **Maintains communication with sponsor, leads documentation, assisted in application design.** |

**1. Introduction**

The purpose of this project will be to develop a cross platform mobile application for Lee Virtual School (LVS). The mobile app will act as a hub for all things LVS. The goal is to have a centralized location for information and resources, similar to their current web page. The project will be developed for both iOS and Android mobile platforms. Users on these platforms will utilize the app to apply to be a student of LVS, track their attendance, view a calendar of events and more. While using secure protocols this app will communicate with databases hosted by LVS to allow in app functionality as opposed to just a shell of a mobile web page. In summary, the goal of this project is to create a mobile app that teachers and students alike will gladly utilize.

**2. Software Requirements Specification**

**2.1 Physical Components**

A user will interact with the mobile application from a mobile device. The application will need access to the Internet for any feature to connect to an appropriate database hosted by LVS.

Submitting a Lee Virtual Instruction Program Application Form will need a connection to the Lee Virtual’s Applications Database. As of now, it is understood that the data a user enters to apply will be sent to the database by the use of a RESTful web service [2]. The Lee County School District’s main agile development team has agreed to host a temporary database on their servers for use during the mobile app’s development. The temporary database and main database can be merged later.

The app will also have access for a user to login for their daily attendance through the Lee County School District’s website. Clicking on a button in the app will bring the login page up in a WebView. No native app capabilities for login can be pursued now due to security purposes. The District’s website will connect to the Lee Virtual Attendance Database as already provided.

A Google Calendar will be implemented in the mobile app to display event information. This will need the Google Calendar API to communicate with Google’s Databases. Doing so will retrieve the needed event data that will populate in the mobile app. The overall interaction of the mobile app with the school systems is shown in Figure 1.

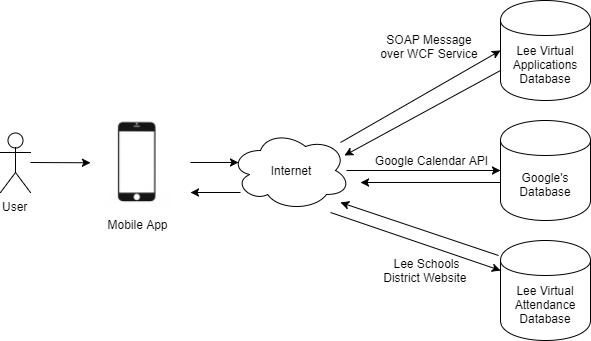


Figure 1 Physical Diagram

**2.2 Software To Be Developed**

The mobile app will target five features to be accomplished: the Calendar, Attendance Login, Staff Contact Information, Resource Links, and an Application Form. Though some Input/output interactions are shown in the Physical Diagram, the Context Diagram as shown in Figure 2 covers more of the user interactions.

After the Google Calendar API populates the events, a user can click on an event to see more information and to perform more actions. A user will click a button to be directed to the attendance website. Any interaction with this website itself is not in the input/output information because that software has already been developed and will not be modified by the mobile app. The Staff Contact Information and Resource Links are information that will be viewable by the user and can be navigated through general user actions. The Application form can be filled out by the user and submitted to the Lee Virtual Applications Database via a RESTful web service.. The detailed information to what is in this form can be found in the Design Description.

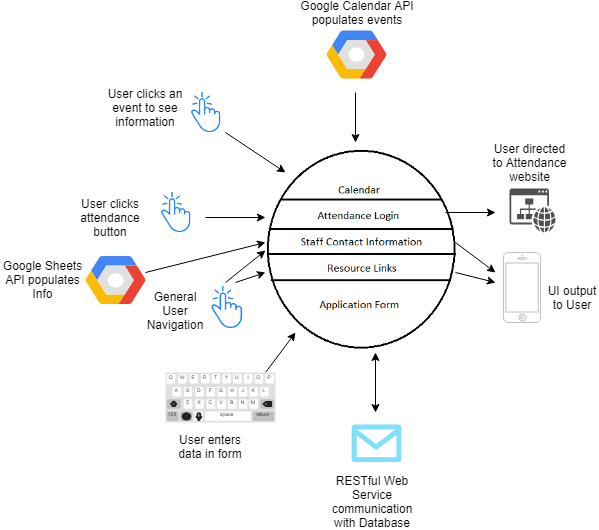


Figure 2 Context Diagram for Software

**2.3 Software Requirements**

The following is a list of software requirements for the project.

1. The software shall function as a mobile app for iOS and Android devices.
2. The software shall implement a way to view events from a Google Calendar within the application.
3. The software may optionally implement a way for a user to add an event from the school’s Google Calendar into their personal Google Calendar.
4. The software shall provide a link for a student to login for their daily attendance on the District’s website.
5. The software shall provide the school’s staff contact information including (but not limited to) Email, Phone, Available Hours, Courses, and a link to the Genbook appointment scheduling website.
6. The software shall provide a list of resource links for students.
7. The software shall provide a “native” form to submit applications into Lee Virtual that includes all sections listed in the current PDF version of the application as shown in the Appendix LVS Application PDF.
8. The software shall submit the application form to Lee Virtual School’s Applications Database via a RESTful web service upon completion of an application form by the user and requesting the submission.

**3. Design Description**

Figure 3 separates the mobile app into the five target objectives: The Calendar, Attendance Login, Staff Contact Information, Resource Links, and an Application Form. The previous Context Diagram (Fig. 2) describes the communication with the user and databases outside of the software circle.

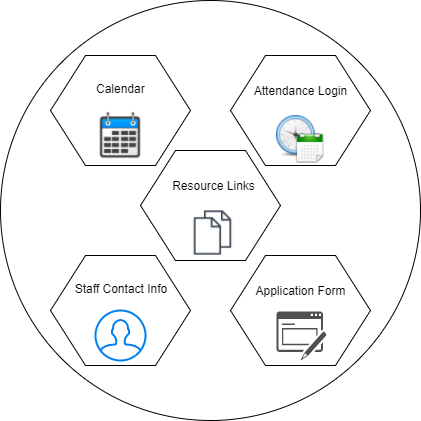


Figure 3 Software Architecture

**3.1 The Calendar**

The school’s public Google calendar should be accessible without authenticating the client library in the Google API since the calendar is public. All that will be needed is the creation of a Calendar Service with the API key and Application Name created in the Google Developers Console. The logic in loading the Calendar Page in the app is shown in Figure 4. The GUI designs for the Calendar Page are shown in Figures 5 and 6.

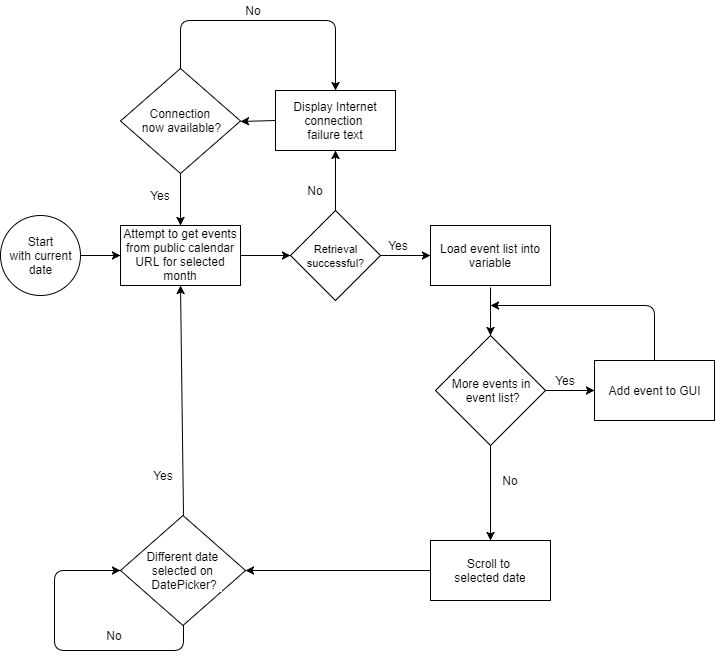


Figure 4 Calendar Page Flowchart

Initially, the Calendar page will display the current month’s events as a list and auto scroll to today’s date. Additionally, as shown in Figure 5, the DatePicker in the top left corner will allow the user to select a different date to be displayed. The ScrollView only shows the events for the month selected and auto scrolls to the day selected.

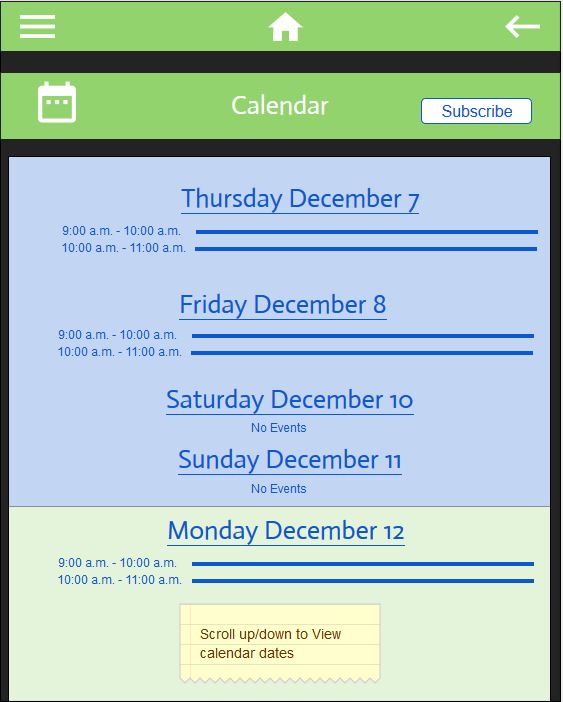


Figure 5 Calendar GUI Initial

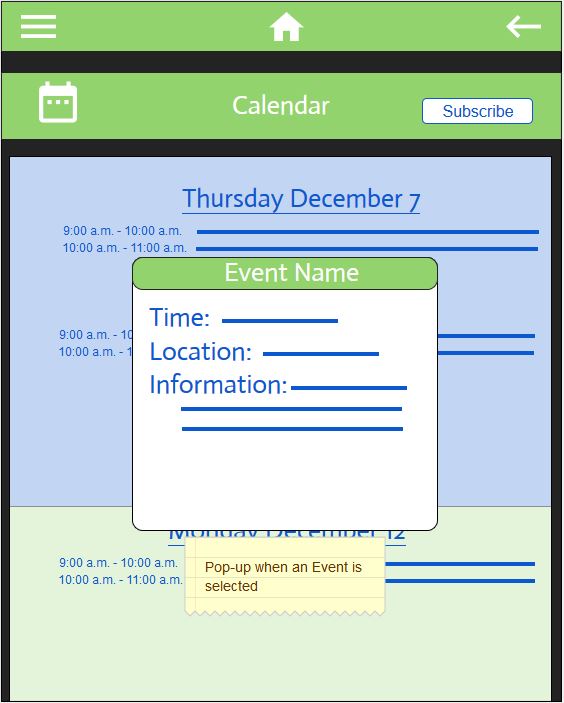


Figure 6 Calendar GUI Event Info Popup

When an event is tapped on, a popup will be displayed with more detailed event information for the user to view. If the optional requirement No. 3 is implemented, the user should also be able to add the selected event to their own personal Google calendar.

For a user to subscribe to the school’s public Google calendar, the user must perform steps in their phone settings; this action cannot be automated. Therefore, when a user clicks the “Subscribe” button in the top right of the Calendar page, a tutorial popup will be shown. Depending on which mobile device the user is currently using, a series of images will appear in a Content Selector box for the user to swipe through to see the tutorial on how to subscribe.

**3.2 Attendance Login**

Daily Attendance is required for every LVS student. When clicking the “Daily Login” menu item in the main menu, a WebView will be displayed with the Lee Schools District website as shown in Figure 7.

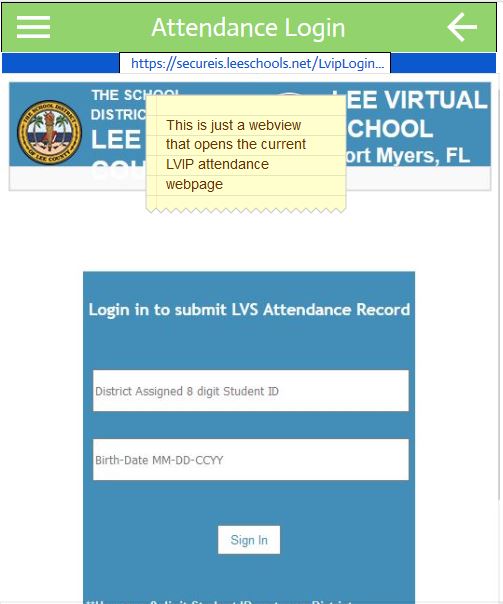


Figure 7 Attendance Login GUI

The native Web Browser on the user’s device will be in charge of displaying an error message if an Internet connection was not established (Figure 8).

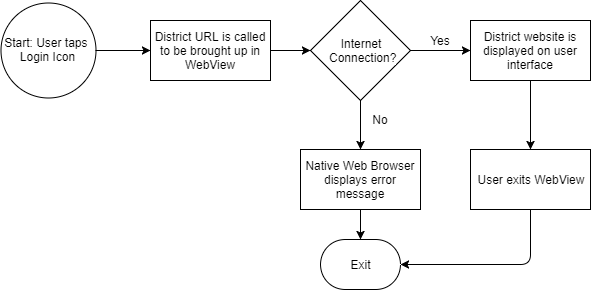


Figure 8 Login Flowchart

**3.3 Contact Information**

In the main menu of the app, the “Teachers” menu item will extend further to more menu options of “Elementary” and “Secondary” when tapped (Figure 9). When one of these options is selected, the proper list of teachers will be displayed to the user (Figure 10). When a specific teacher is selected, all contact information for that teacher will be display to the user (Figure 11).

Currently, the design calls for the contact information to be hard coded into the mobile application. Further expansion of the project may call for this information to be read and interpreted from an outside source.

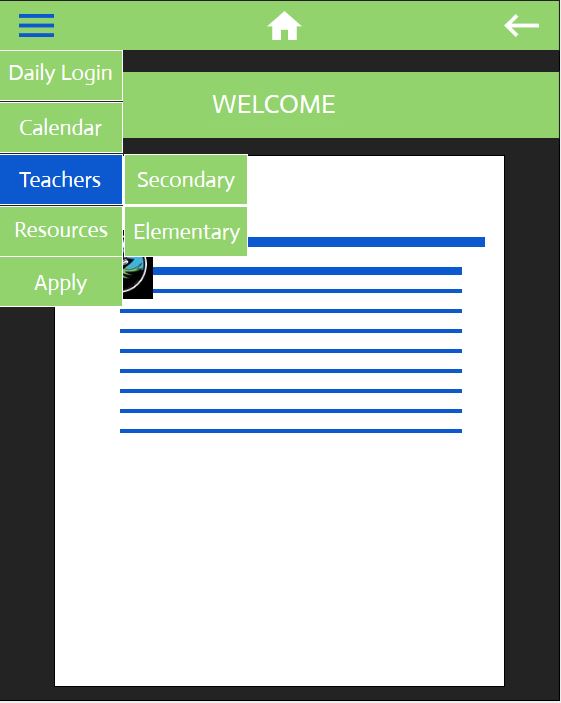


Figure 9 Main Menu “Teacher Extension” GUI

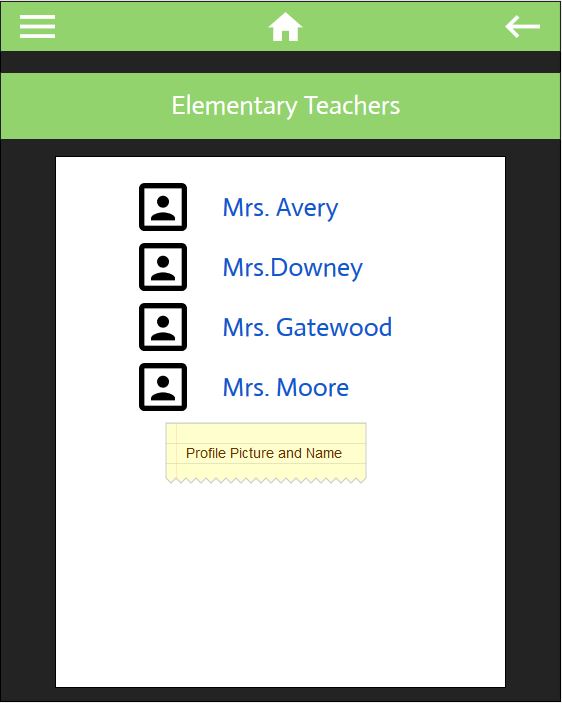


Figure 10 Teacher List GUI - Elementary Example

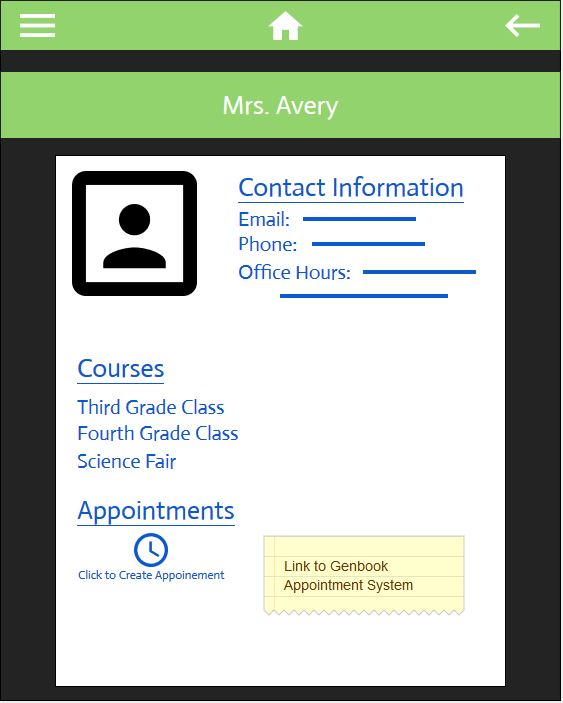


Figure 11 Teacher’s Contact Information GUI

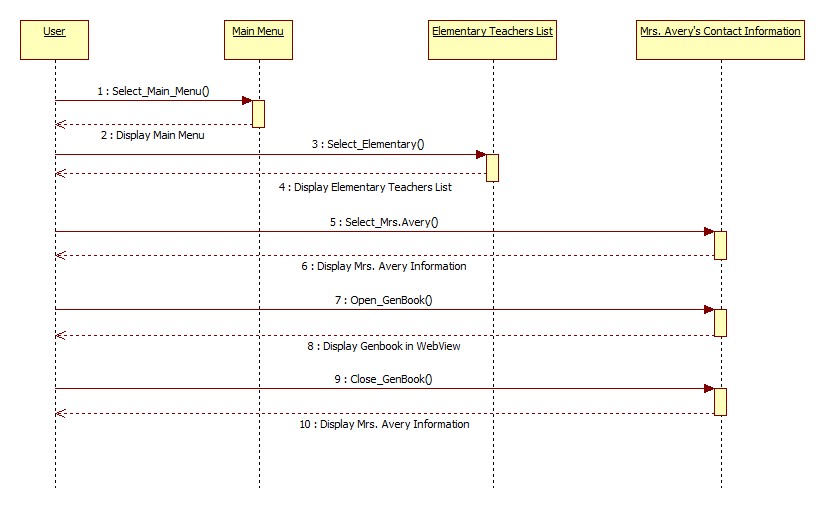


Figure 12 Contact Information Sequence Diagram

Figure 12 shows a sequence diagram of an example situation where a user may want to view Mrs. Avery’s (an elementary teacher) information and make an appointment. The user will go through the main menu to be directed to the teacher list and then to Mrs. Avery’s Contact Page. In the Contact Page, a link will be available to schedule an appointment through GenBook via WebView.

**3.4 Resource Links**

Students will be able to see a list of resource links after selecting the “Resources” menu item in the main menu (Figure 13). All links are navigation links to other basic content pages in the app for the specified resource content including text, images, and other links. Any links in these resource content pages will direct the user to their device’s native Internet Browser, not a WebView within the app. Content will be based off the LVS website under the Resources section.



Figure 13 Resources Page GUI

**3.5 Application Form**

The native application form will be shown when “Apply” is selected from the main menu. The application form will gather all data needed as in the PDF version of the form (see Appendix). The pages will be divided into Student Information (Figure 14), School Information (Figure 15), Guardian Information (Figure 16), and Additional Questions (Figure 17).

For the first three pages, a “Next” button will be displayed at the bottom to navigate to the next page. However, for the last page (Additional Questions), a “Submit” button will be displayed at the bottom of the page to submit the application to the LVS Applications Server. Currently, the team is requesting more information and access to the LVS Application Server.

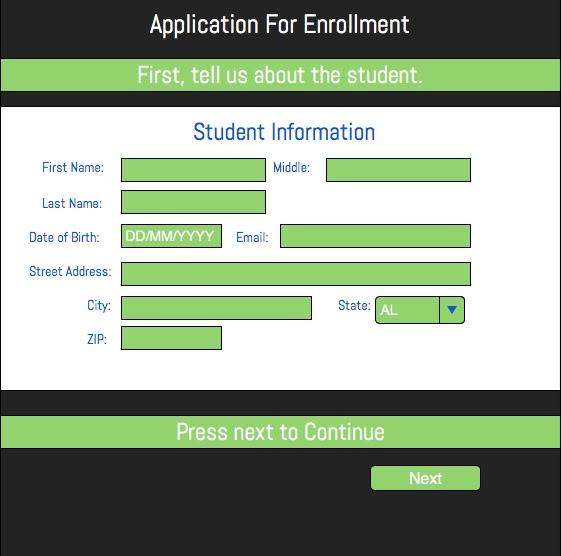


Figure 14 Application Form GUI - Student Information

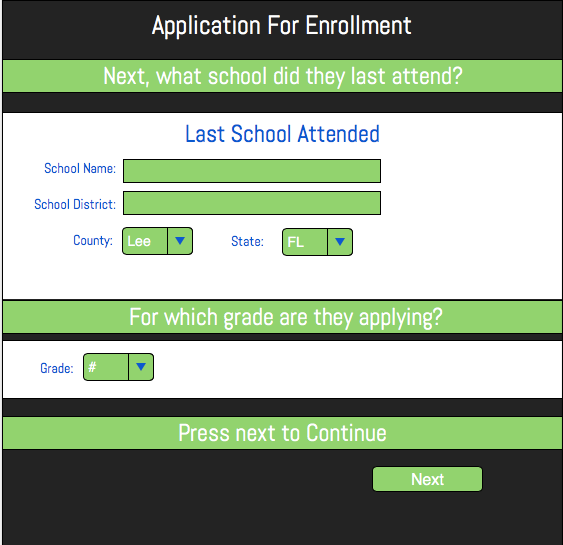


Figure 15 Application Form GUI - School Information

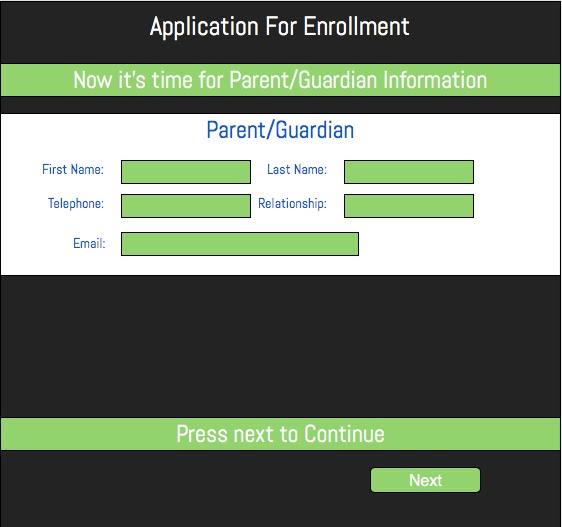


Figure 16 Application Form GUI - Guardian Information

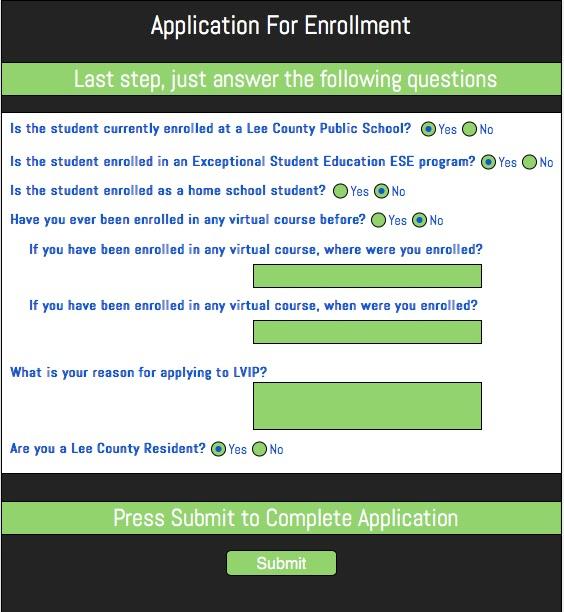


Figure 17 Application Form GUI - Additional Questions

**3.6 Sequence of Actions**

The software will perform proper validation on the necessary fields on the current page when the “Next” or “Submit” button is tapped. An Application Form will only be submitted after all fields are properly validated and if a connection to the server is established (Figure 18).

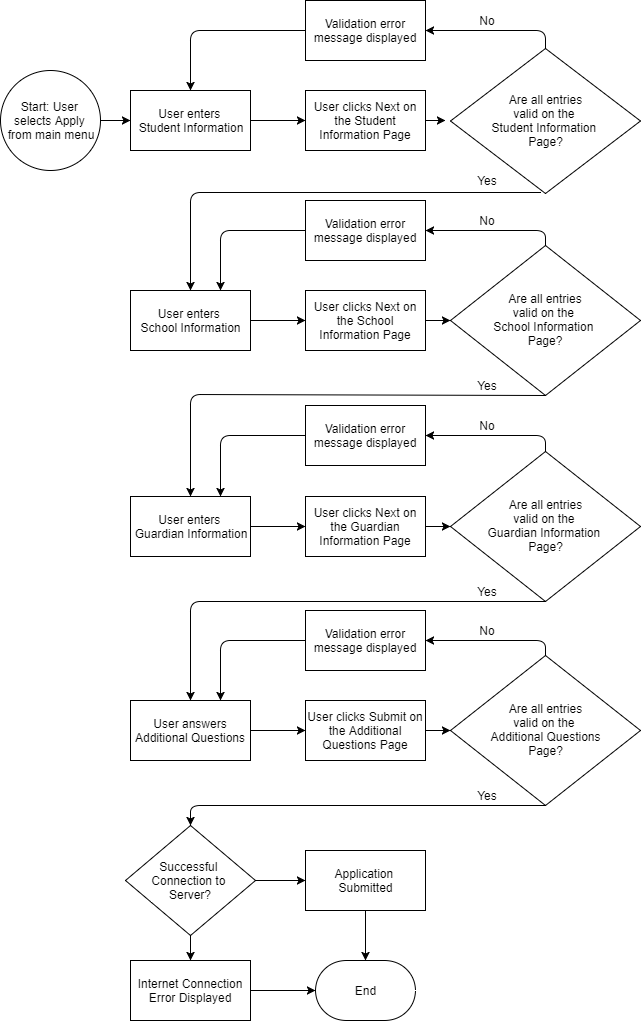
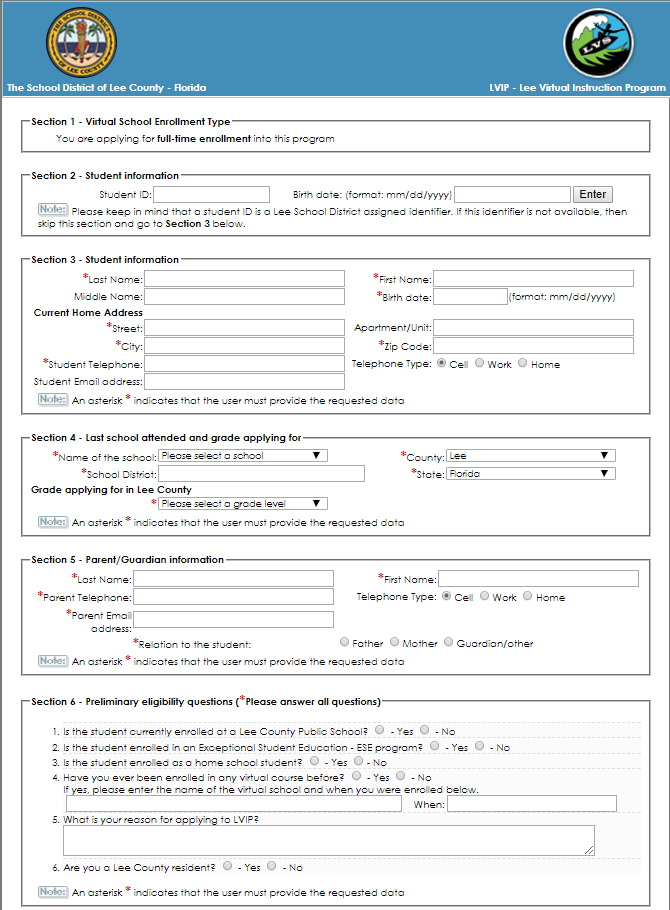
****

Figure 18 Flowchart for Submitting an Application Form

**References**

1. Microsoft Corp.,“What Is Windows Communication Foundation.” Microsoft Docs, 30 March 2017, <https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>
2. Microsoft Corp., “Consuming a RESTful Web Service”, May 2017, <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/data-cloud/consuming/rest>

**Appendix - LVS Application PDF**

****

**Appendix - Meetings with Sponsors**

1/23/18

3:30 PM - 4:15 PM  
Those Present: Shelby Smith, Derek Bailey, Al Shilling (Sponsor)  
Topics Discussed:  
 The sponsor approved our design document and shared it amongst his organization.  
After gathering input from other stakeholders within the organization, a small list of possible feature additions was created. He discussed the reason and need for each new feature. He understood that we intend to at least deliver the minimum viable product and can look into adding features after the minimum viable product is solid.

2/27/18

3:00 PM - 3:30 PM

Those Present: Shelby Smith, Derek Bailey, Al Shilling (Sponsor), Jason Stanford (Software Engineer)

Topics Discussed:

The Team conducted a demo of the project prototype to the Sponsors and received approval with the design. The Team requested needed resources of vector-based logo files and a test database connection. Mr. Stanford and Derek collaborated on the formatting of a JSON object to submit to the database.

4/3/18

3:00 PM - 3:20 PM

Those Present: Shelby Smith, Derek Bailey, Al Shilling (Sponsor), Jason Stanford (Software Engineer)

Topics Discussed:

Approval was received to continue using the third-party SyncFusion Schedule to display the data from the Google calendar. A user manual was submitted to the Sponsors to complete before April 20th to integrate their own Google Calendar and Google Sheets into the app for delivery. Mr. Stanford reported the delay in the database connection due to firewall changes and will be contacting us with the database connection within a week. Mr. Shilling will also be submitting vector-based logo images this week. The Sponsors were notified of the ETI Presentations taking place on April 27th.