*Florida International University*

*School of Computing and Information Sciences*

# 

# CIS 4911 - Senior Project

Software Engineering Focus

Final Deliverable

Project Title: Go Local Staff 3.0

**Team Members:**

Daylin Gonzalez

**Product Owner(s)**:

Eduardo Garcia

**Mentor(s)**:

Mohsen Taheri

**Instructor**:

Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*This document presents the information necessary to gain a good understanding of the Go Local Staff 3.0 project. The Go Local Staff is a web site that connects employers looking to host events with staff willing to work at those events. In the following pages, user stories, diagrams and information will be presented that will describe the project in detail.*

**Table of Contents**

**Introduction** ……………………………...…………………………………………………………… 5

Current System ………………………………………………………………………………………..5 Purpose of New System …………………………………………………………………………... 5

**User Stories**

Implemented User Stories ……………………..……………………………………………………..6

Pending User Stories …………………………………………………………………..………..….. 9

**Project Plan**

Hardware and Software Resources ……………………………………………………………….… 10

Sprints Plan …………………………………………………………………………………………. 12

*Sprint 1*  …………………………………………………………………………………………... 12

*Sprint 2*  …………………………………………………………………………………………... 13

*Sprint 3*  …………………………………………………………………………………………... 13

*Sprint 4*  …………………………………………………………………………………………... 13

*Sprint 5* ……………………………………………………………………………………………. 14

*Sprint 6*  …….……………………………………………………………………………………... 14

**System Design**

Architectural Patterns ..…………………………………………………………………………….. 16 System and Subsystem Decomposition ………………………………………….………………….17

Deployment Diagram ……………………………………………………………………….……….18

Design Patterns ………………………………………………………………………………….…...18

**System Validation**  ……………………….…………………………………………………………….19

**Glossary**  ……….……………………………………………………………………………………….38

**Appendix**  ……………………………………………………………………………………………….39

Appendix A - UML Diagrams ………………………..……………………………………………. 39

*Static UML Diagrams* …………………………………………………………………………….*39*

*Dynamic UML Diagrams*  ……………………………………………………………………..….42

Appendix B - User Interface Design ………………………………………………………….…......44

Appendix C - Sprint Review Report ...………………………………………………………...…… 47

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents …………………………………………………………….……………………………...51

**References** ………………………………………………………………………...………... **………..**54

**Introduction**

In this section, the current system and the proposed system will be described. This section is meant to establish the scope and relevance of the project by providing the reader with a brief overview of the problem that this application is meant to address. The sections following this one will show the users stories worked on, the project planning that done, the system design that was laid out, and the validation tests that were carried out.

## 

## Current System

In order to find and manage the required staff to host events, employers have to go through a difficult and resource intensive process. Most of the times this process involves a middlemen which incur in high overhead costs as a result. Communication and management are also bottlenecked by having to consult with these middlemen in order to accomplish anything

Staffing agencies attempt to tackle this problem by providing access to their collection of available talent. However, this collection is limited, often resulting in the need to visit several of these agencies in order to execute an event.

The process of connecting talent to employers involves the use of email, phone and fax in order to exchange the necessary information. The staffing agencies compile all of the information and communication into their computer systems and use said systems to manage the entire process.

## Purpose of New System

Go Local Staff is a web page that aims to solve these issues by eliminating the middleman. The web page allows staff to register by entering personal information, and employers to search for staff by the criteria needed for a specific event.

This system removes the bottlenecks of staffing agencies by connecting employers directly to their staff. User only need to search for the staff they are interested in and the web site will provide the necessary information to fulfill the user’s needs.

# User Stories

The following section provides the detailed user stories that were implemented in this iteration of the Go Local Staff project. These user stories served as the basis for the implementation of the project’s features. This section also shows the user stories that are to be considered for future development.

## Implemented User Stories

**Use Case #866 – Review website**

· As a developer, I need to see if the website previously developed is usable, so that it can be further developed or if it should be started from scratch

Acceptance Criteria

* A decision is made

**Use Case** #**879 – Research Squarespace**

· As a developer, I need to research Squarespace to know if it is suitable for our web page

Acceptance Criteria

* A decision is made

**User Story #880 - Create a registration page so the users can register in our web page**

* As a developer I need to create a registration page, users can register in our web page
* As a developer, I need to connect to the server to see what is inside so I can develop the registration page
* As a developer, I need to connect to the current database to analyze the current tables design
* As a developer I need to create a user so the database can be accessed remotely
* As a developer, I have to install Asp.Net Core local and in server to be able to publish web page into the server because there are compatibility issues

Acceptance Criteria

* Registration page is created

**Use Story** #**885 – Modify registration page**

* As a developer I need to modify the registration page so users can register in a sequence of steps
* As a developer I need to connect the registration page to the current database

Acceptance Criteria

* Registration page is divided in different pages
* Registration Page is connected to the current database

**User Story #895 - Add picture and video upload options to the registration page**

* As a developer, I need to add pictures and videos upload options so users can upload pictures or videos to their profiles

Acceptance Criteria

* Users can upload pictures
* Users can upload videos

**User Story #896 - Send Registration Page to the server**

* As a developer, I need to send the registration page to the server so users can open it from their computers and start creating accounts

Acceptance Criteria

* Registration page is available in the web

**User Story #900 - Add optional information message to registration page**

* As a developer, I have to add a message to the registration page so the user knows what information is optional

Acceptance Criteria

* Registration page shows a message in the optional information part

**User Story #907 - Create HTML/PHP database query page for website**

* As a developer, I have to implement a database query page for the current website so administrator can see what is in the database

Acceptance Criteria

* Database query page is implemented

## 

## 

## 

## 

## 

## 

## 

## Pending User Stories

**User Story #887 - Change domain name of current web page**

* As a developer I need to change the domain name of the current web page

Acceptance Criteria

* The domain name is changed

# 

# 

# 

# 

# 

# 

# Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

## Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

**Digital Ocean**

Digital Ocean is the provider used to host the website and server containing the database used in the application. Their service was chosen because of the compelling price and effective management tools.

**Mingle**

Mingle was used as a planning and management tool for the various agile development processes.

**MySQL**

MySQL was chosen as the relational database because of the open source nature and the group’s familiarity with it.

**Asp.Net Core**

Asp.Net Core was the frame work used to develop the Registration web page because it allows compatibility between all operating systems.

**C Sharp**

C Sharp was used as the server-side language for the Registration web page development. C Sharp was chosen because it is the most suitable language for ASP.Net Core.

**PHP**

PHP was used as the server-side scripting language for the Database Query web page development. PHP was chosen to allow compatibility with the previous website.

**MySQL Workbench**

MySQL Workbench was used to manage the MySQL databases.

**Github**

Github was used to store and manage the source code.

**Gmail**

Gmail was used for communication.

**Google Drive**

Google Drive was used to store project documents and to transfer data between the group members.

**Visual Studio 2015 IDE**

Visual Studio 2015 was used as the IDE in which the website was developed. Visual Studio 2015 was chosen because it is developed and supported by Microsoft and is the go-to solution for Asp.Net development.

**Nginx**

Nginx was used as the web server for the Registration web page. Nginx was chosen because it can be configured as a reverse proxy.

## 

## Sprints Plan

### Sprint 1

(08/27/2016 - 09/09/2016)

***User Story*** *#****866 – Review website***

· Review what is done in the website

***Acceptance Criteria***

· A decision is made

***Modeling***

No modeling needed for this user story

### Sprint 2

(9/10/2016 - 09/23/2016)

***User Story*** *#****873 – Setup development environment***

· Research and install software needed

***Acceptance Criteria***

· Environment is set up

***Modeling***

No modeling needed for this user story

### Sprint 3

(09/24/2016 - 10/07/2016)

***User Story*** *#****879 – Research Squarespace***

· Learn how Squarespace works

***Acceptance Criteria***

· A decision is made

***Modeling***

No modeling needed for this user story

***User Story*** *#****880 – Create Registration Page***

· Install Asp.Net Core

· Create user for remote access to database

· Connect to current database to analyze tables

· Connect to server

· Create Registration Page

***Acceptance Criteria***

· Registration page is created

***Modeling***

Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

### Sprint 4

(10/8/2016 - 10/21/2016)

***User Story*** *#****885 – Modify registration page***

· Modify registration page

· Connect registration page to database

***Acceptance Criteria***

· Registration page is divided in different pages

· Registration page is connected to the database

***Modeling***

Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

***User Story*** *#****887 – Change domain name of current web page***

· Change domain name of current web page

***Acceptance Criteria***

· Domain name is changed

***Modeling***

No modeling needed for this user story

### Sprint 5

(10/22/2016 - 11/4/2016)

***User Story*** *#****895 – Add picture and video upload options to the registration page***

· Add video upload option to the registration page

· Add picture upload option to the registration page

***Acceptance Criteria***

· Users can upload pictures

· Users can upload videos

***Modeling***

Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

***User Story*** *#****896 – Send Registration Page to the server***

· Send Registration Page to the server

***Acceptance Criteria***

· Registration page is available in the web

***Modeling***

No modeling needed for this user story

### Sprint 6

(11/5/2016 - 11/11/2016)

***User Story*** *#****900 – Add optional information message to registration page***

· Add optional information message to registration page

***Acceptance Criteria***

· Registration page shows a message in the optional information part

***Modeling***

Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

***User Story*** *#****907 – Create HTML/PHP database query page for website***

· Create a search page

· Create Login webpage for the Administrator

***Acceptance Criteria***

· Database query page is implemented

***Modeling***

Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

# 

# System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

## Architectural Patterns

The main architectural pattern used for this project is Model View Controller (MVC). MVC was chosen because it was the pattern that best fit website application development. The IDE Visual Studio 2015 in general is set up to use the MVC pattern when development web pages and using said pattern allows us to use Visual Studio to its fullest potential. In addition to MVC, the system also uses a client-server architecture wherein the client is the website and the server is the Ubuntu LAMP server, where all of the client’s requests are made, that is hosted on Digital Ocean.

## 

## 

## 

## 

## 

## System and Subsystem Decomposition

The diagram below, Figure 1, shows the subsystem decomposition of our system. The system is divided into four parts. The first is the view which is essentially the user interface that the user interacts with.

The next part of our system is the controller subsystem. This subsystem is the link between the views the user sees and the data that is in the models.

The third subsystem is the model. The model are the classes needed to interact with the database. The controller subsystem communicates with these classes in order to access or update data that is on the database.

The last part of the system is the database itself. This database is on the Ubuntu server that is hosted on Digital Ocean’s servers. This database is what the model connects to in order to access data.

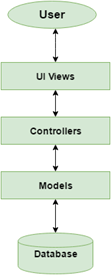


Figure 1- Subsystem Decomposition

## 

## Deployment Diagram

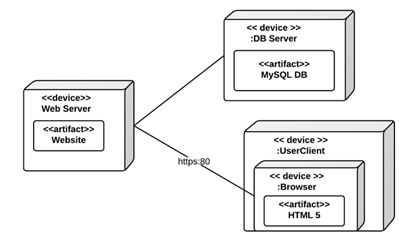


Figure 2 – Deployment Diagram

## Design Patterns

The design pattern used in the system is the Front controller pattern. The Front controller pattern provides a centralized entry point for handling requests.

# 

# 

# 

# System Validation

In this section, all of the test cases that were down to validate our system are outlined. All of the testing was done manually, following the input requirements of the test cases.

**Unit Test**

|  |
| --- |
| **Test ID: UT-001** |
| **Purpose: Test that the SignUp() in the SignUpController class returns the registration page view** |
| **Preconditions: Internet browser is open and application is running** |
| **Input: //localhost:60097/Home/SignUp** |
| **Expected Output: The registration page shows up with all the required fields** |

|  |
| --- |
| **Test ID: UT-002** |
| **Purpose: Test that the Add(entry) and SaveChanges() methods**  **in the OurDatabaseContext class updated the database** |
| **Preconditions: Application is running** |
| **Input: new registered\_staff() { StaffID = 85, FirstName = "John", LastName = "Winston" };** |
| **Expected Output: The new registered\_staff is added to the database** |

|  |
| --- |
| **Test ID: UT-003** |
| **Purpose: Test that the View(view) method**  **in the SignUpController class returned the view specified** |
| **Preconditions: Application is running** |
| **Input: View(“RequiredInfo”);** |
| **Expected Output: The new RequiredInfo view is displayed** |

|  |
| --- |
| **Test ID: UT-004** |
| **Purpose: Test that the View(view) method**  **in the RequiredInfoController class returned the view specified** |
| **Preconditions: Application is running** |
| **Input: View(“AdditionalInfo”);** |
| **Expected Output: The new AdditionalInfo view is displayed** |

|  |
| --- |
| **Test ID: UT-005** |
| **Purpose: Test that the View(view) method**  **in the AdditionalInfoController class returned the view specified** |
| **Preconditions: Application is running** |
| **Input: View(“AccountCreated”);** |
| **Expected Output: The new AccountCreated view is displayed** |

|  |
| --- |
| **Test ID: UT-006** |
| **Purpose: Test that the SignUp() method**  **in the SignUpController class created a new account** |
| **Preconditions: Application is running and model state is valid** |
| **Input: SignUp(email, password, confirmPassword);** |
| **Expected Output: The new account is created** |

|  |
| --- |
| **Test ID: UT-007** |
| **Purpose: Test that the RequiredInfo() method**  **in the RequiredInfoController class added info to the account** |
| **Preconditions: Application is running and model state is valid** |
| **Input: RequiredInfo(FirstName,MiddleInitial,LastName,Address,City,ZipCode,State,Phone,DateOfBirth,Gender );** |
| **Expected Output: The new account is updated with the new info** |

|  |
| --- |
| **Test ID: UT-008** |
| **Purpose: Test that the AdditionalInfo() method**  **in the AdditionalInfoController class added info to the account and returned AccountCreated view** |
| **Preconditions: Application is running and model state is valid** |
| **Input: AdditionalInfo(StaffType,NativeLanguage,SecondLanguage,ThirdLanguage,TypeDL,Ethnicity,Height,Weight,HairColor,EyeColor,ShirtSize,PantSize,ChestSize,WaistSize,HipSize, DressSize,ShoeSize,Tattoos,Piercings,DesiredHourlyRate,DesiredWeeklyRate,SsnOrEin, BusinessName,Travel,Insurance, BankRouting, AccountNumber );** |
| **Expected Output: The new account is updated with the new info and AccountCreated view is returned** |

|  |
| --- |
| **Test ID: UT-009** |
| **Purpose: Test that the RequiredInfo(entry) method**  **in the RequiredInfoController class saves the picture to a local folder and saves the information in the database** |
| **Preconditions: Application is running, user uploads a picture and clicks Next button** |
| **Input: 3.jpg** |
| **Expected Output: The new picture is saved to a local folder and the information is saved in the database** |

|  |
| --- |
| **Test ID: UT-010** |
| **Purpose: Test that the RequiredInfo(entry) method**  **in the RequiredInfoController class validates that the picture file is in a valid format** |
| **Preconditions: Application is running, user uploads a picture and clicks Next button** |
| **Input: 3.mov** |
| **Expected Output: The page displays a message : "Please choose either a JPG, TIF, GIF or PNG image."** |

|  |
| --- |
| **Test ID: UT-011** |
| **Purpose: Test that the RequiredInfo(entry) method**  **in the RequiredInfoController class creates a folder named as the staff id inside the uploads/pictures local folder to save that user’s picture** |
| **Preconditions: Application is running, user uploads a picture and clicks Next button** |
| **Input: 3.gif** |
| **Expected Output: The folder uploads/pictures/1235/3.gif is created** |

|  |
| --- |
| **Test ID: UT-012** |
| **Purpose: Test that the RequiredInfo(entry) method**  **in the RequiredInfoController class validates that the user uploads a picture** |
| **Preconditions: Application is running and user clicks Next button** |
| **Input:** |
| **Expected Output: The page shows a message "Field can't be empty"** |

|  |
| --- |
| **Test ID: UT-013** |
| **Purpose: Test that the AdditionalInfo(entry) method**  **in the AdditionalInfoController class saves the video to a local folder and saves the information in the database** |
| **Preconditions: Application is running, user uploads a video and clicks Submit button** |
| **Input: 3.mov** |
| **Expected Output: The new video is saved to a local folder and the information is saved in the database** |

|  |
| --- |
| **Test ID: UT-014** |
| **Purpose: Test that the AdditionalInfo(entry) method**  **in the AdditionalInfoController class validates that the video file is in a valid format** |
| **Preconditions: Application is running, user uploads a video and clicks Submit button** |
| **Input: 3.jpg** |
| **Expected Output: The page displays a message : "Please choose either a AVI, FLV, WMV, MOV, or MP4 video."** |

|  |
| --- |
| **Test ID: UT-015** |
| **Purpose: Test that the AdditionalInfo(entry) method**  **in the AdditionalInfoController class creates a folder named as the staff id inside the uploads/videos local folder to save that user’s video** |
| **Preconditions: Application is running, user uploads a video and clicks Submit button** |
| **Input: 3.mov** |
| **Expected Output: The folder uploads/pictures/1236/3.mov is created** |

|  |
| --- |
| **Test ID: UT-016** |
| **Purpose: Test that the AdditionalInfo(entry) method**  **in the AdditionalInfoController class validates that the Resume file is in a valid format** |
| **Preconditions: Application is running, user uploads a resume and clicks Submit button** |
| **Input: resume.txt** |
| **Expected Output: The page displays a message : "Please choose either a DOC, DOCX, TXT, PDF, or RTF document."** |

|  |
| --- |
| **Test ID: UT-017** |
| **Purpose: Test that the AdditionalInfo(entry) method**  **in the AdditionalInfoController class creates a folder named as the staff id inside the uploads/resumes local folder to save that user’s resume** |
| **Preconditions: Application is running, user uploads a resume and clicks Submit button** |
| **Input: resume.docx** |
| **Expected Output: The folder uploads/pictures/1248/resume.docx is created** |

|  |
| --- |
| **Test ID: UT-018** |
| **Purpose: Test that the Config.php class connects to the database** |
| **Preconditions: Application is running** |
| **Input: Username: daylin; Password: password** |
| **Expected Output: The Search page is shown** |

|  |
| --- |
| **Test ID: UT-019** |
| **Purpose: Test that the Session.php class creates a session after validating username and password** |
| **Preconditions: Application is running and user enters username and password** |
| **Input: Username: daylin; Password: password** |
| **Expected Output: The Search page is shown showing the name of the user** |

|  |
| --- |
| **Test ID: UT-020** |
| **Purpose: Test that the Index page validates the username and password** |
| **Preconditions: Application is running** |
| **Input: Username: daylin; Password: password** |
| **Expected Output: The Search page is shown with the name of the user** |

|  |
| --- |
| **Test ID: UT-021** |
| **Purpose: Test that the Index page validates the username and password** |
| **Preconditions: Application is running** |
| **Input: Username: daylin123; Password: password** |
| **Expected Output: The Index page shows a message "Your Login Name or Password is invalid"** |

|  |
| --- |
| **Test ID: UT-022** |
| **Purpose: Test that the CreateQuery function of the Search page validates creates the correct query** |
| **Preconditions: Application is running, user enters data and press Submit** |
| **Input: {Sort By: First Name; Order: Ascending; First Name: “”; Last Name: Gonzalez; Gender: Either; Has Photo: “”; Has Video: “”; No Tattoos: “”; No Piercings: “”; Email: “”; Phone: “”; Height: at most “”; Weight: at most “”; Hair Color: is SELECT; Eye Color: is SELECT; Chest: “”-””; Waist: “”-””; Hips: “”-””; Shirt Size: SELECT; Pant Size: SELECT; Dress Size: SELECT; Shoe Size: SELECT; Ethnicity: “”; Languages: “”; Promotional Sectors: “”; Locations: “”; Will Travel: Yes}** |
| **Expected Output: "SELECT \* FROM registered\_staff WHERE lastName = \'gonzalez\' and travel = \'Yes\' ORDER BY firstName ASC"** |

|  |
| --- |
| **Test ID: UT-023** |
| **Purpose: Test that the CreateQuery function of the Search page validates creates the correct query** |
| **Preconditions: Application is running, user enters data and press Submit** |
| **Input: {Sort By: First Name; Order: Descending; First Name: “Daylin”; Last Name: Gonzalez; Gender: Either; Has Photo: Yes; Has Video: “”; No Tattoos: “”; No Piercings: “Yes”; Email: “”; Phone: “”; Height: at most “”; Weight: at most “”; Hair Color: is SELECT; Eye Color: is SELECT; Chest: “”-””; Waist: “”-””; Hips: “”-””; Shirt Size: SELECT; Pant Size: SELECT; Dress Size: SELECT; Shoe Size: SELECT; Ethnicity: “”; Languages: “Spanish”; Promotional Sectors: “”; Locations: “”; Will Travel: “”}** |
| **Expected Output: "SELECT \* FROM registered\_staff WHERE firstName = \'Daylin\' and lastName = \'Gonzalez\' and imageName != \'null\' and piercings = \'no\' and (nativeLanguage = \'Spanish\' or secondLanguage = \'Spanish\' or thirdLanguage = \'Spanish\') ORDER BY firstName DESC"** |

|  |
| --- |
| **Test ID: UT-024** |
| **Purpose: Test that the CreateQuery function of the Search page validates creates the correct query** |
| **Preconditions: Application is running, user enters data and press Submit** |
| **Input: {Sort By: Last Name; Order: Descending; First Name: “”; Last Name: “”; Gender: Female; Has Photo: Yes; Has Video: Yes; No Tattoos: “”; No Piercings: “Yes”; Email: “”; Phone: “”; Height: at most 5’6”; Weight: at most “”; Hair Color: is SELECT; Eye Color: is SELECT; Chest: “”-””; Waist: “”-””; Hips: “”-””; Shirt Size: SELECT; Pant Size: SELECT; Dress Size: SELECT; Shoe Size: SELECT; Ethnicity: “”; Languages: “Spanish”; Promotional Sectors: DJ; Locations: Kansas; Will Travel: “”}** |
| **Expected Output: "SELECT \* FROM registered\_staff WHERE gender = \'Female\' and imageName != \'null\' and videoName != \'null\' and piercings = \'no\' and height <=5.6 and height > 0.0 and (nativeLanguage = \'Spanish\' or secondLanguage = \'Spanish\' or thirdLanguage = \'Spanish\') and staffType = \'DJ\' and state = \'KS\' ORDER BY lastName DESC"** |

|  |
| --- |
| **Test ID: UT-025** |
| **Purpose: Test that the Search page shows a message when no data matches the criteria** |
| **Preconditions: Application is running, user enters data and press Submit** |
| **Input: {Sort By: Last Name; Order: Descending; First Name: “”; Last Name: “”; Gender: Female; Has Photo: Yes; Has Video: Yes; No Tattoos: “”; No Piercings: “Yes”; Email: “”; Phone: “”; Height: at most 5’6”; Weight: at most “”; Hair Color: is SELECT; Eye Color: is SELECT; Chest: “”-””; Waist: “”-””; Hips: “”-””; Shirt Size: SELECT; Pant Size: SELECT; Dress Size: SELECT; Shoe Size: SELECT; Ethnicity: “”; Languages: “Spanish”; Promotional Sectors: DJ; Locations: Kansas; Will Travel: “”}** |
| **Expected Output: “No data matches the criteria”** |

|  |
| --- |
| **Test ID: UT-026** |
| **Purpose: Test that the LogOut page destroys the session** |
| **Preconditions: Application is running, user presses Sign Out link** |
| **Input:** |
| **Expected Output: Index page is shown** |

|  |
| --- |
| **Test ID: UT-027** |
| **Purpose: Test that the Search\_Submit page shows the data retrieved** |
| **Preconditions: Application is running, user enters data and press Submit** |
| **Input: {Sort By: First Name; Order: Ascending; First Name: “”; Last Name: “”; Gender: Either; Has Photo: “”; Has Video: “”; No Tattoos: “”; No Piercings: “”; Email: “”; Phone: “”; Height: at most “”; Weight: at most “”; Hair Color: is SELECT; Eye Color: is SELECT; Chest: “”-””; Waist: “”-””; Hips: “”-””; Shirt Size: SELECT; Pant Size: SELECT; Dress Size: SELECT; Shoe Size: SELECT; Ethnicity: “”; Languages: “”; Promotional Sectors: “”; Locations: “”; Will Travel: Yes}** |
| **Expected Output: System shows all the data** |

|  |
| --- |
| **Test ID: UT-028** |
| **Purpose: Test that the Search\_Submit page shows the data retrieved** |
| **Preconditions: Application is running, user enters data and press Submit** |
| **Input: {Sort By: First Name; Order: Ascending; First Name: “”; Last Name: “Gonzalez”; Gender: Either; Has Photo: “”; Has Video: “”; No Tattoos: “”; No Piercings: “”; Email: “”; Phone: “”; Height: at most “”; Weight: at most “”; Hair Color: is SELECT; Eye Color: is SELECT; Chest: “”-””; Waist: “”-””; Hips: “”-””; Shirt Size: SELECT; Pant Size: SELECT; Dress Size: SELECT; Shoe Size: SELECT; Ethnicity: “”; Languages: “”; Promotional Sectors: “”; Locations: “”; Will Travel: Yes}** |
| **Expected Output: System shows all the data with Last Name = Gonzalez in descending order of First Name** |

**Integration Test**

|  |
| --- |
| **Test ID: IT-001** |
| **Purpose: Verify whether the web page will have all the required fields to create an account** |
| **Preconditions: Internet browser is open and application is running** |
| **Input: http://localhost:60097/Home/SignUp** |
| **Expected Output: The registration page shows up with all the required fields** |

|  |
| --- |
| **Test ID: IT-002** |
| **Purpose: Verify whether the user is able to create an account** |
| **Preconditions: Internet browser is open and application is running** |
| **Input: { Email = daylin2512@yahoo.es, FirstName = "John", MiddleInitial = “”, LastName = "Winston", Address = “40 Sidonia Ave”, City = “Coral Gables”, ZipCode = “33134”, State = “FL”, Phone = “7865518745”, DateOfBirth = “12/15/85”, Gender = “Female”, StaffType = “Dancer”, NativeLanguage = “Spanish”, SecondLanguage = “English”, ThirdLanguage = “”, TypeDL = “”, Ethnicity = “”, Height = “5’6””, Weight = “130”, HairColor = “Black”, EyeColor = “Green”, ShirtSize = “S”, PantSize = “27”, ChestSize = “”, WaistSize = “”, HipSize = “”, DressSize = “M”, ShoeSize = “ 37”, Tattoos = “No”, Piercings = “Yes”, DesiredHourlyRate = “”, DesiredWeeklyRate = “”, SsnOrEin = “458785412”, BusinessName = “”, Travel = “ Yes”, Insurance = “No”, BankRouting = “”, AccountNumber = “” }** |
| **Expected Output: The account is created and the confirmation page shows up** |

|  |
| --- |
| **Test ID: IT-003** |
| **Purpose: Verify whether the user is able to upload a picture and a video** |
| **Preconditions: Internet browser is open and application is running** |
| **Input: { Email = daylin25125@yahoo.es, FirstName = "John", MiddleInitial = “”, LastName = "Winston", Address = “40 Sidonia Ave”, City = “Coral Gables”, ZipCode = “33134”, State = “FL”, Phone = “7865518745”, DateOfBirth = “12/15/85”, Gender = “Female”, Picture = “pic.jpg”, StaffType = “Dancer”, NativeLanguage = “Spanish”, SecondLanguage = “English”, ThirdLanguage = “”, TypeDL = “”, Ethnicity = “”, Height = “5’6””, Weight = “130”, HairColor = “Black”, EyeColor = “Green”, ShirtSize = “S”, PantSize = “27”, ChestSize = “”, WaistSize = “”, HipSize = “”, DressSize = “M”, ShoeSize = “ 37”, Tattoos = “No”, Piercings = “Yes”, DesiredHourlyRate = “”, DesiredWeeklyRate = “”, SsnOrEin = “458785412”, BusinessName = “”, Travel = “ Yes”, Insurance = “No”, BankRouting = “1234567890”, AccountNumber = “123456789”, Video = “video1.mov”, Resume = “resume.docx” }** |
| **Expected Output: The account is created and the confirmation page shows up** |

|  |
| --- |
| **Test ID: IT-004** |
| **Purpose: Verify whether the user is able to see the optional information message** |
| **Preconditions: Internet browser is open and application is running** |
| **Input: { Email = daylin25125@yahoo.es, FirstName = "Daylin", MiddleInitial = “M”, LastName = "Gonzalez", Address = “40 Sidonia Ave”, City = “Coral Gables”, ZipCode = “33134”, State = “FL”, Phone = “7865518745”, DateOfBirth = “12/15/85”, Gender = “Female”, Picture = “pic.jpg” }** |
| **Expected Output: The system displays the additional information page with a message on top informing the user that the information is not required but may increase the chances to get get a job.** |

|  |
| --- |
| **Test ID: IT-006** |
| **Purpose: Verify whether the user is able to log in** |
| **Preconditions: Application is running, user enters username and password press Submit** |
| **Input: Username: daylin; Password: password** |
| **Expected Output: The Search page is shown with the name of the user** |

|  |
| --- |
| **Test ID: IT-007** |
| **Purpose: Verify whether the user is able to search the database using criteria** |
| **Preconditions: Application is running, user enters data and press Submit** |
| **Input: {Sort By: Last Name; Order: Ascending; First Name: “”; Last Name: “”; Gender: “”; Has Photo: “”; Has Video: “”; No Tattoos: “”; No Piercings: “”; Email: “”; Phone: “”; Height: at most “”; Weight: at most “”; Hair Color: is SELECT; Eye Color: is SELECT; Chest: “”-””; Waist: “”-””; Hips: “”-””; Shirt Size: SELECT; Pant Size: SELECT; Dress Size: SELECT; Shoe Size: SELECT; Ethnicity: “”; Languages: “”; Promotional Sectors: “”; Locations: “”; Will Travel: “”}** |
| **Expected Output: System shows all the data** |

|  |
| --- |
| **Test ID: IT-008** |
| **Purpose: Verify whether the user is able to search the database using criteria** |
| **Preconditions: Application is running, user enters data and press Submit** |
| **Input: {Sort By: Last Name; Order: Ascending; First Name: “”; Last Name: “”; Gender: Female; Has Photo: Yes; Has Video: Yes; No Tattoos: “”; No Piercings: Yes; Email: “”; Phone: “”; Height: at most “”; Weight: at most “”; Hair Color: is SELECT; Eye Color: is SELECT; Chest: “”-””; Waist: “”-””; Hips: “”-””; Shirt Size: SELECT; Pant Size: SELECT; Dress Size: SELECT; Shoe Size: SELECT; Ethnicity: “”; Languages: “”; Promotional Sectors: “”; Locations: FL; Will Travel: “”}** |
| **Expected Output: System shows all the data with Gender = Female, Has Photo = Yes, Has Video: Yes, No Piercings: Yes, Locations: FL** |

|  |
| --- |
| **Test ID: IT-009** |
| **Purpose: Verify whether the user is able to log out** |
| **Preconditions: Application is running, user logged in press Sign Out** |
| **Input:** |
| **Expected Output: The index page is shown** |

**Glossary**

**LAMP**: LAMP is an acronym for the web service stack containing a Linux based OS, an Apache web server, a MySQL database system, and PHP code.

**Asp.Net Core:** Asp.Net Coreis a free and open-source web framework, and the next generation of ASP.NET, developed by Microsoft and the community. It is a modular framework that runs on both the full .NET Framework, on Windows, and the cross-platform .NET Core.

**Nginx:** Nginx is a software to provide a web server. It can act as a reverse proxy server for TCP, UDP, HTTP, HTTPS, SMTP, POP3, and IMAP protocols, as well as a load balancer and an HTTP cache.

# 

# 

# 

# 

# 

# Appendix

## Appendix A - UML Diagrams

*Static UML Diagrams*

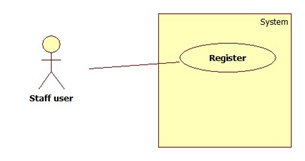


Figure A1 – Use case diagram for registration page

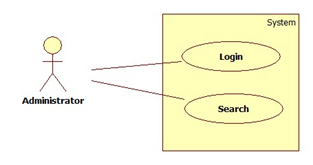


Figure A2 – Use case diagram for Database Query page

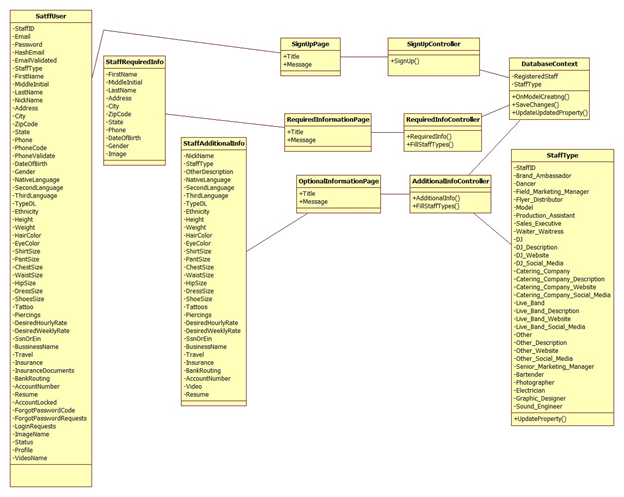


Figure A3 – Class diagram for Registration page

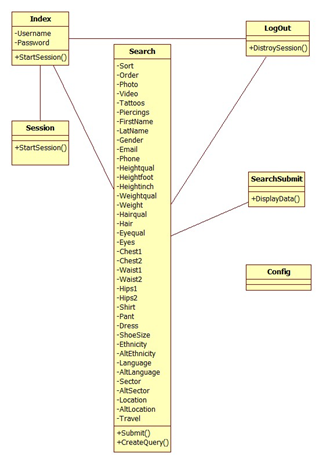


Figure A4 – Class diagram for Database Query page

*Dynamic UML Diagrams*

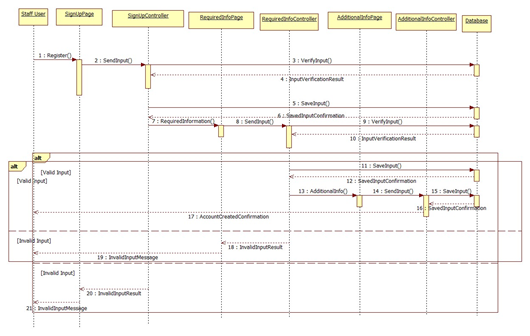


Figure A5 – Sequence diagram for Registration page

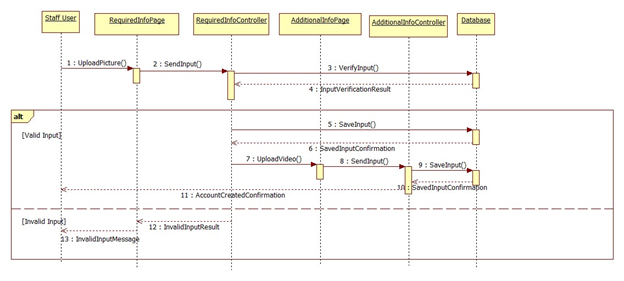


Figure A6 – Sequence diagram for Registration page picture and video upload

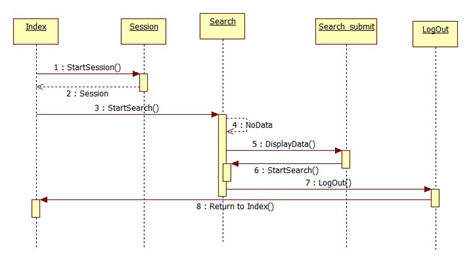


Figure A7 – Sequence diagram for Database Query page

## 

## 

## 

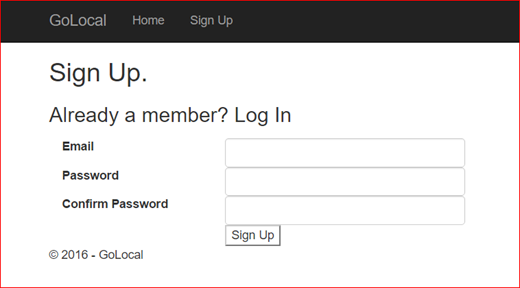
## 

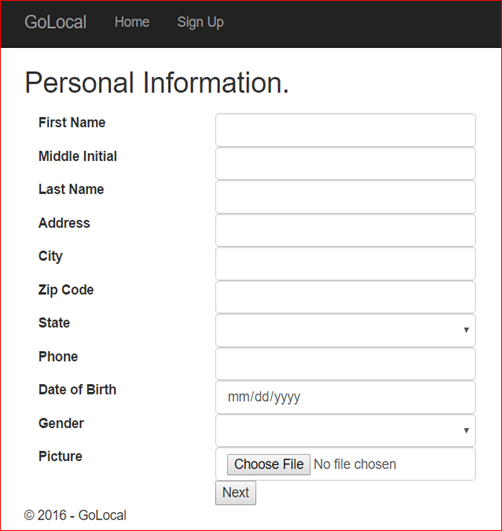
## 

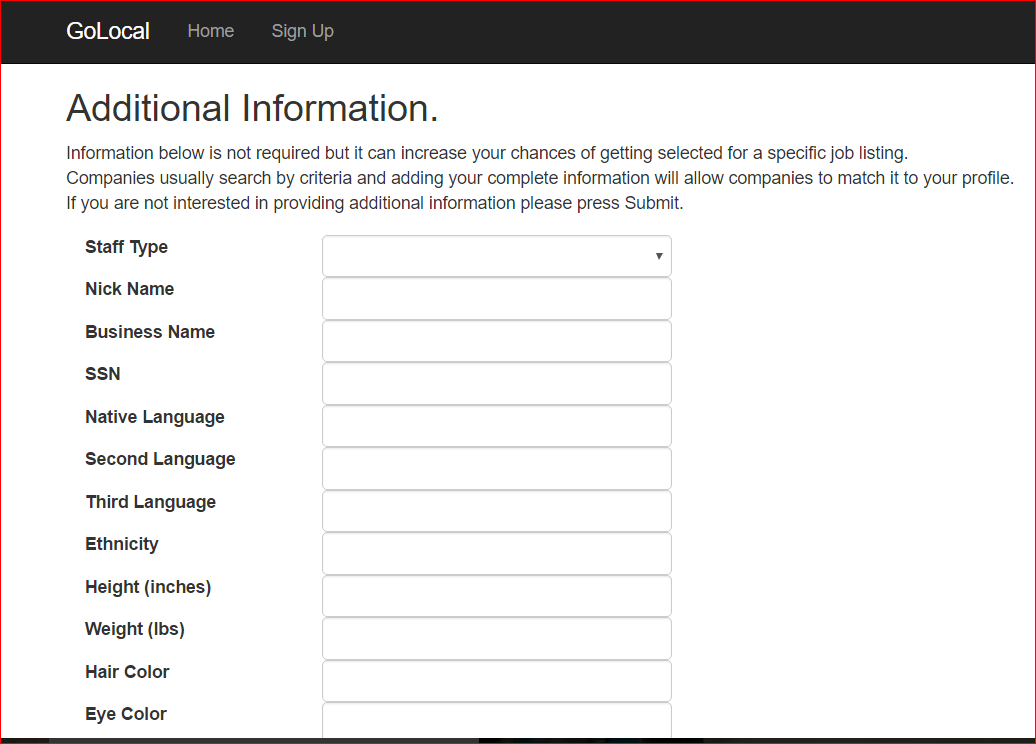
## 

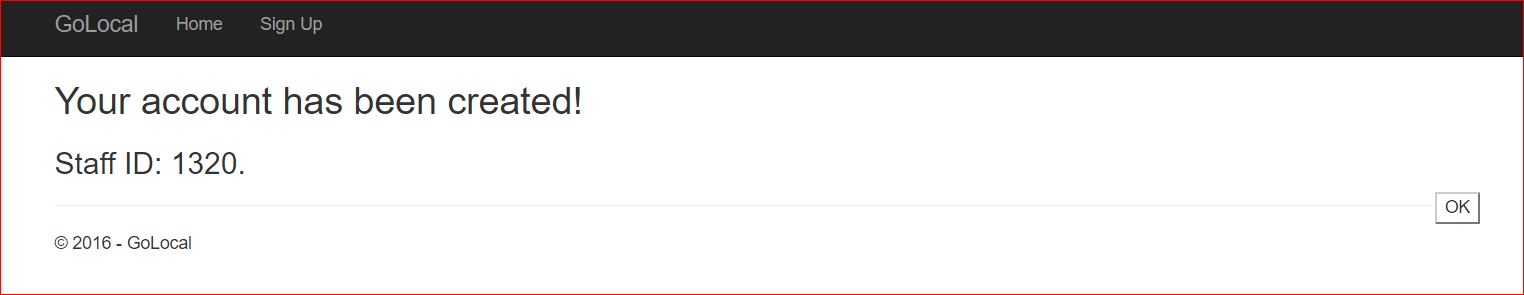
## 

## Appendix B - User Interface Design

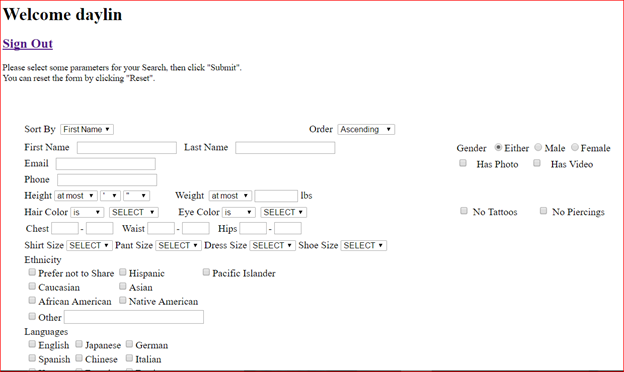












## 

## 

## 

## 

## 

## 

## 

## Appendix C - Sprint Review Reports

**Sprint 1**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel, Gabriel Espinales

Start time: 8/31/16 7pm

End time: 8/31/16 8:30pm

After discussion, the velocity of the team were estimated to be 30.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

1. User Story
2. Review previously developed website for decisions
3. Review App installation option
4. See if Registration Page can be linked to other website

The team members indicated their willingness to work on the following user stories.

* Gabriel Espinales

· User Story 2.Review app

* Daylin Gonzalez Miguel

· User Story 1.Review website

· User Story 3.Registration Page

**Sprint 2**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel, Gabriel Espinales

Start time: 9/10/16 8:30pm

End time: 9/10/16 9:30pm

After discussion, the velocity of the team were estimated to be 30

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

1. User Story

1. Set up environment to start web page from scratch

2. Change web page domain and structure

The team members indicated their willingness to work on the following user stories.

* Gabriel Espinales

· User Story 2.Change web page domain and structure

* Daylin Gonzalez Miguel

· User Story 1.Setup development environment

**Sprint 3**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel, Gabriel Espinales

Start time: 9/23/16 9:00pm

End time: 9/23/16 10:00pm

After discussion, the velocity of the team were estimated to be 30

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

1. User Story

1. Research Squarespace
2. Create Registration page

The team members indicated their willingness to work on the following user stories.

* Gabriel Espinales

· User Story 1.Research Squarespace

* Daylin Gonzalez Miguel

· User Story 1.Research Squarespace

· User Story 2.Create registration page

**Sprint 4**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel

Start time: 10/10/16 9:00pm

End time: 10/10/16 10:00pm

After discussion, the velocity of the team were estimated to be 30

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

· User Story

1. Modify Registration page
2. Change domain name of current web page

The team members indicated their willingness to work on the following user stories.

* Daylin Gonzalez Miguel

· User Story 1.Modify registration page

· User Story 2. Change domain name of current web page

**Sprint 5**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel

Start time: 10/22/16 2:00pm

End time: 10/22/16 3:00pm

After discussion, the velocity of the team were estimated to be 30

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

· User Story

1. Add picture and video upload options to registration page
2. Publish registration page in the server
3. Change domain name of current web page

The team members indicated their willingness to work on the following user stories.

* Daylin Gonzalez Miguel

· User Story 1.Add picture and video upload options to registration page

· User Story 2. Publish registration page in the server

· User Story 3. Change domain name of current web page

**Sprint 6**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel

Start time: 11/5/16 8:30pm

End time: 11/5/16 9:30pm

After discussion, the velocity of the team were estimated to be 30.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

· User Story

1. Add optional information message to registration page
2. Create HTML/PHP database query page for website

The team members indicated their willingness to work on the following user stories.

* Daylin Gonzalez Miguel

· User Story 1.Add optional information message to registration page

· User Story 2. Create HTML/PHP database query page for website

## 

## 

## Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

**Sprint reviews**

**Sprint 1**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel, Gabriel Espinales

Start time: 9/10/16 7:30pm

End time: 9/10/16 8:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 1.Review website
* User Story 2.Review app
* User Story 3.Registration Page

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* None

**Sprint 2**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel, Gabriel Espinales

Start time: 9/23/16 8:00pm

End time: 9/23/16 8:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 1.Setup development environment
* User Story 2.Change web page domain

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* None

**Sprint 3**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel

Start time: 10/10/16 7:30pm

End time: 10/10/16 8:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 1.Research Squarespace
* User Story 2.Create Registration Page

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* None

**Sprint 4**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel

Start time: 10/22/16 1:00pm

End time: 10/22/16 1:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 1.Modify Registration page

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* User Story 2.Change domain name of current web page

**Sprint 5**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel

Start time: 11/5/16 7:30pm

End time: 11/5/16 8:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 1.Add picture and video upload options to the registration page
* User Story 2.Send registration page to the server

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* User Story 2.Change domain name of current web page

**Sprint 6**

Attendees: Eddie Garcia, Daylin Gonzalez Miguel

Start time: 11/19/16 7:30pm

End time: 11/19/16 8:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 1.Add optional information message to registration page
* User Story 2.Create HTML/PHP database query page for website

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

# 

# References

The work of the previous group, that is, the Go Local App2.0 group was used in the making of this report.