**C868 – Software Capstone Project Summary**

**Task 2 – Section A**



|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | Playas Limpias – A Volunteer-Based Coast Recue Application |
| **Student Name:** | Victor J., Olivo Ramirez |

# Table of Contents

[**Business Problem** 3](#_Toc1937181)

[**The Customer** 3](#_Toc1937182)

[**Business Case** 3](#_Toc1937184)

[**Fulfillment** 4](#_Toc1937185)

[**Existing Gaps** 5](#_Toc1937186)

[**SDLC Methodology** 6](#_Toc1937187)

[**Deliverables** 8](#_Toc1937188)

[**Project Deliverables** 8](#_Toc1937189)

[**Product Deliverables** 10](#_Toc1937190)

[**Implementation** 11](#_Toc1937191)

[**Validation and Verification** 12](#_Toc1937192)

[**Environments and Costs** 12](#_Toc1937193)

[**Programming Environment** 12](#_Toc1937194)

[**Environment Costs** 13](#_Toc1937195)

[**Human Resource Requirements** 14](#_Toc1937196)

[**Project Timeline** 14](#_Toc1937197)

# **Business Problem**

## **The Customer**

The customer is the Puerto Rico Department of Natural and Environmental Resources (Departamento de Recursos Naturales y Ambientales de Puerto Rico, DRNA for short), a non-profit government entity located in the Caribbean Island Puerto Rico. DRNA currently does not have a dedicated platform for coast cleaning events and rely on third-party non-profit organizations to take the initiative and perform these types of events. Many of these organizations are founded by locals concerned about the increasing problem of coast lines contamination. The island of Puerto Rico has many of the most beautiful coast lines in the world, consequently these coast lines received high volume of visitors throughout the year, even higher on holidays. Many of the visitors bring multiple food, drinks, and many other items that contain packaging plastics and other materials. Sadly, it has become an increasing problem that many of the visitors leave these items behind contaminating the beautiful coastlines and posing risks to many maritime animals. It's our duty to protect our coastlines, every year millions of pounds of trash are left at beaches by both locals and tourists. When all this trash stays behind in the shorelines it can be dragged into the ocean putting many more ecosystems at risk. ‘’Marine litter is not only ugly – it can harm ocean ecosystems, wildlife, and humans. It can injure coral reefs and bottom-dwelling species and entangle or drown ocean wildlife. Some marine animals ingest smaller plastic particles and choke or starve” (marinelittersolutions.com, What is marine debris? 2020).

Many individuals will gladly volunteer their time and help clean our coastlines, but to make a significant impact, many volunteers is often needed. Many non-profit organizations perform coast rescue events (trash clean up), however they mostly do on an annual basis, which is not enough to fight the pollution on our coastlines. These organizations provide some sort of coordination and create events that make some positive impact.

In addition to the fact that these events are performed only once a year, on most occasions, volunteer participation in such events is low. Volunteers for these events are gathered mostly through social media and word-of-mouth, which is inefficient. When a possible participant sees a post of an event or hears it from someone, they will likely forget the date and place of the event in the next 5 minutes, unless they actively set some sort of reminder.

## 

## **Business Case**

Playas Limpias – the proposed software solution is a volunteer-based coast rescue web application. This web application will be specifically designed to help DRNA increase their reach for volunteers and with the main objective of helping shorelines stay clean and consequently preserving many ecosystems. At the moment, DRNA does not possess any type of platform to organize public events that align with the organization’s mission which is to protect and conserve natural resources of the island. DRNA currently rely on and incentivize external organizations that take the initiative and perform beach cleanup events to rescue severe affected areas. These organizations have a huge positive impact, but it is not enough to combat shoreline contamination. The events are not frequent, and their participation is often for employees and not always open to the public. The island of Puerto Rico has so many beautiful coastlines and the local people are proud and willing to do more to preserve such beautiful ecosystems. DRNA can take advantage of this to achieve their mission of protecting our natural resources, but they need a platform where locals can have the opportunity to volunteer and do their part in conserving our shorelines.

Playas Limpias web application will allow DRNA to extend their reach and have a bigger positive impact on the island natural resources. The web application will provide a range of functionalities and views to inform the website visitors of the purpose and importance of keeping our shorelines clean and allow them to be part of a community that is making a positive impact on the environment. The website will receive the user with an informational landing page, this home page will contain information like purpose, objective, contact information and a navigation bar for users to navigate to the other views of the web application. Navigating through the web application, users will be able to see all current events happening around the island, all their details, and provide the ability for the user to volunteer and set reminders for events of their choosing. Users, after registering successfully, will also have the ability to create their own custom events providing the opportunity for other users to volunteer and to join them in rescuing the proposed shoreline in need. When creating an event, the user will be prompted to provide essential information such as name and date for the event, estimated number of volunteers required, location, and optionally upload an image of the affected area.

# **SDLC Methodology**

The delivery of the Playas Limpias web application software solution will follow a Modified Waterfall methodology. The waterfall model is a predictive methodology that consists of multiple phases that must be followed in a specific order. Every phase must be completed in its entirety before the project can move on to the next phase. There are some variations of the waterfall methodology but in its plain version, waterfall methodology does not allow to go back to a previous phase once the project has moved on to the next. However, the modified waterfall model allows for a feedback system in which we can go back to the previous phase for verification and validation. Delivering Playas Limpias web application using a waterfall model allows to determinate a clear end-state of the final product, which permits to set clear deadlines, more accurate cost estimates, and the product completion date. The modified waterfall model is perfectly suited for delivering Playas Limpias software solution in view that the customer problem is very well understood and the software requirements clear.

First step in the chosen methodology is the requirements phase. In this phase all the software solution requirements are define and agreed upon between the customer and the solutions team. The project delivery team will ensure all the stated requirements are clear, detailed and completely understood by everyone. At the end of this phase a requirements document will be produced and delivered to all stakeholders. Also, major deadlines, estimated cost and project completion date are result of this phase.

Next, we move onto the design phase, in this phase the project team will design the complete solution from top to bottom. Using the requirements documents as a guide, all the needed technologies and tools required to deliver the proposed solution will be defined here. As a result, from this phase, we will have three major documents that will support the rest of the solution development:

* Low-Fidelity Wireframe
* Database Diagram
* Testing Plan

The next step is the implementation phase. In this phase, hence its name, we will implement all that has been planed and agreed in the previous phases. We will use all the previous documents as a guide to develop the actual code and design of the software solution. Here the development team will work with multiple technologies and programing languages to bring the software solution to life. At the end of this phase, we would have produced a final software solution prototype that complies with all the previous stated requirements on its entirety.

Now that we have a final product, we move on onto the testing phase. Here, we perform various tests for quality assurance. No code ever has been 100% bug-free, testing is the best offense to provide the best quality possible in the final software solution. In this phase we will test the application substantially looking for any deviation from the requirements, design flaws, or any logic errors. In every occasion an error is discovered it will be immediately addressed and fix.

After various tests, iterations, and only when the quality standard has been met, we will move onto the delivery and deployment phase. In this phase we will move the application into its final destination, the production environment. Once the complete software solution is successfully deployed into the production environment it can be accessed and use by the customer and its clients.

The last step is the maintenance phase. In this phase we will ensure a proper handoff of the application to the customer. The project team will discuss and inform the customer with all the necessary information to maintain the application running without any major setbacks. Also, it will be discussed plans of action for new discovered bugs, on-going costs for maintaining the application, and extend a maintenance service option.

# **Deliverables**

In this section we will discuss in more detail all the previously mention documents and deliverables that will be produced through the creation of the Playas Limpias software solution. Here, we outlined and describe each deliverable in a more clear and concise matter.

## **Project Deliverables**

* Requirements document
  + Contains all the requirements that the final software solution must meet in order to be considered completed.
  + It must be agreed upon by both the customer and the project team.
  + It must be clear and unambiguous.
  + It should include details such as, estimated costs, project scope, limitations, and a high-level project timeline.
* Project Schedule
  + Here we breakdown into more detail the high-level project timeline from the requirements document.
  + A clearer view of the project timeline is presented and includes the following:
    - Start date
    - Major milestones deadlines
    - Deliverables deadlines
    - Phases durations
    - Scheduled time off
    - Planned make-up days
    - Anticipated project competition date
* Low-fidelity Wireframe
  + The low-fidelity wireframe will present a blueprint for the project team to create the user interface design and give a clear overview of the application flow.
* Database Diagram
  + The database schema representing different entities will be presented here
  + It will include data types and relations between the different database tables
* Testing Plan
  + The testing plan will be subdivided into two parts:
    - Unit Tests: Input Validations, Individual Methods, and Error handling
    - Functionality Test: A quality assurance tester will perform a series of defined steps to simulate user functionality. Each test plan will contain all the detailed steps to replicate, input data and desired output expectations. A selected combination of these tests will serve as the customer acceptance test.
  + In the event of a test failure, the assessed section of code will be evaluated for potential bugs and fix.

## **Product Deliverables**

* Landing page with information about the web application (customer) objective and mission
* Custom URL
* Constant navigation bar to navigate through the web application
* User interface design and flow that matches the produced wireframe
* Functional database that matches with the database schema and suffice the application storage needs
* A finalized software solution that meets all the requirements agreed upon
* Fully deploy the application to a secure hosting cloud service provider
* Provided initial data

# **Implementation**

Implementation of the software solution will not require any interruption of DRNA operations in view that the proposed software solution is not replacing or interacting with any of the existing services provided by the organization. During the requirements phase data and information recollection will be necessary for the development of the requirements document. The recollection of information can happen via a few short meetings or, alternatively via email communications. After all the necessary information and detail are collected, a final requirements document will be produced, review by both parties and agreed upon. This document will serve as the basis for the development of the entire software solution and will also be made available to all stakeholders.

Once the requirements phase is completed, the project team will require minimum to none interaction with DRNA since all the necessary information in regards to the application is clearly defined in the produced document. Moving on, and always using the requirements document as guide, the project team will move on to design the user interface, wireframe, database schema, and determined all the technologies and tools that will be required for the creation of the product. After all the design is finalized, the team will start programing the application into life.

Lastly, after thorough testing to meet quality standards, Playas Limpias web application will be fully deployed and hosted into the cloud and a maintenance plan discussed with the customer. The deployment to the cloud is anticipated to go smoothly, is cost-effective and will provide elasticity for the future of the application.

# **Validation and Verification**

A comprehensive testing plan will be developed by the project team to ensure the functionality, look, and feel of the final product meets all the requirements. There is no such thing as a bug-free code, especially not on the first go, testing is a crucial part of developing software. A quality standard should always be tested against the programing code to be certain is ready for production. It is always more cost effective to dedicate time into finding and fixing bugs earlier, than doing it after the application has been deployed. The earlier the detection the easier and low-costly can be to fix. Also, major undetected bugs can be costly for all involved organizations, both financially and for the reputation. For all these reasons, we take very seriously and understand the importance of quality testing.

During the testing phase, a quality assurance developer will create multiple white-box unit tests verifying different sections of the code that deal with input validation, processing, and error-handling. Every method will be tested against a criteria and results will be come in the form of a ‘passed’ or ‘failed’ score. Each test will be designed with the objective of covering all common and extreme cases of inputs, this will maximize the chance of finding bugs. After all the unit tests have been written and reviewed, all these tests will run automatically against the code and gives us the results in just a brief moment. The QA developer will then carefully review each test result, inspect in detail all the ones that failed, fix any bugs in the evaluated code, and re-run all the tests. This process might take many iterations and the testing phase will not be considered completed until all the encountered bugs have been evaluated, fixed, and all the tests have passed successfully.

In the second part of the testing phase, functionality tests will be created with the objective of simulating user interaction scenarios across the web application. These tests aim to help us confirm the existence and proper functionality of all the requirements defined in the requirements document. These tests are considered black-box tests and will be performed manually by a designated tester. Each test will contain detailed steps to be performed and a summary of the expected output.

# **Environments and Costs**

## **Programming Environment**

Playas Limpias web application will be develop using the ASP.NET Core framework from Microsoft. Most of the development will take place inside the IDE Visual Studio 2022 which includes many of the necessary tools and libraries that will facilitate the development and deployment of the application. The main programing language for the development of this software solution will be C#, but will also be utilizing many others like SQL, HTML, CSS, and JavaScript. For the development we will also be taking advantage of some open-source frameworks like Bootstrap and jQuery, these will help us make the application more responsive and eye-pleasing to the user in a shorter amount of time. For the database, we will be using MySQL, an open-source relational database solution.

For deployment, Playas Limpias will be hosted in the cloud service provider Azure. Visual Studio makes the transition from the programing environment to the cloud almost effortless. Visual Studio IDE provides us with tools that do most of the work automatically and deploy the application to the cloud with a few clicks.

## **Environment Costs**

Azure provides a wide range of services, and the pricing varies according to many factors like frequency of data access, type of web application, combination of services, and many more. When choosing Azure services, it can get overwhelming fast but for this project we have a clear idea of what type of services will be required. Once Playas Limpias have been fully develop, tested, and is ready for deployment, it will require the use of two Azure services, an “App Service” to host the web application and a “SQL Database Service” to host the MySQL database. With so many possibilities and combination of services, it can be easy to miscalculate precise pricing estimates. Luckily for us, Azure provides a “Services Pricing Calculator” that will allow us to simulate the creation of our required services and give us an accurate estimate. One major benefit of using a cloud service provider to host our web application is that we have the ability to pay only what is needed at the moment, meaning we can start with the minimum required processing and storage capacities and increase if needed as the user-base grows. Using the calculator provided by Azure and considering the basic processing and storage options an estimate of $25.32 per month was calculated. For the mentioned price, Playas Limpias web application have all services required to be fully deployed with an “Service Level Agreement” or SLA of 99.99%.

## **Human Resource Requirements**

For the creation of Playas Limpias web application it will require a small team of professionals with different expertise. The team will consist of five experts, one Project Manager (PM), one UI/UX Designer, two Software Developers, and a QA tester. The PM will help the team go through all phases efficiently and on schedule. The PM will also be the main point of communication with the customer, initials requirements and any inquiries from the customer will be filtered from the team by the project manager. Our designer will help us create a visual appealing application and ensure a pleasing user experience. Once the designer paints a complete picture for the team, and following the guidance of the PM, the software developers will take care of bringing the web application to life using different technologies and tools. Finally, we will need our QA tester to perform several tests against the application. The QA tester will work closely with the software developers to ensure optimal quality and fixing any bugs discovered during the testing phase. An estimated cost for assembling this team of experts can be appreciate in the following table:

|  |  |  |
| --- | --- | --- |
| Resource | Rate \* Time | Total |
| Project Manager | $60/h \* 100h | $6000 |
| Designer | $40/h \* 40h | $1600 |
| Software Developers | $70/h \* 80h \* 2 | $11,200 ($5,600 x 2) |
| QA Specialist | $40/h \* 40h | $1600 |
| **TOTAL Human Resource Cost** |  | **$20,400** |

# 

# **Project Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Milestone/Task | Deliverable | Description | Duration |
| Requirements gathering | Meet, discuss, and reach an agreement for the project requirements | Requirements document  Project schedule | PM meets with the customer to reach an agreement regarding the project requirements and create a project schedule. | 1 week |
| System Design | Create low-fidelity wireframe | Low-fidelity website mockup  (Wireframe) | The team’s designer will create and low-fidelity wireframe to translate the visual vision of the project. Once finalized, the PM and developers review with the designer. | 2 days |
| System Design | Create database schema | DB Diagram | The designer with the input of the developers and PM, will also assist in creating the database diagram. | 2 days |
| System Design | Create testing plan | Testing Plan | The PM, developers and QA tester get together to design several efficient and thoroughly tests. | 3 days |
| Implementation | Create the web application and unit tests | First final main version of the Web application | The developers will develop the full application according to the requirements and the PM guidance. All unit tests will be developed. | 2 weeks |
| Testing Phase | Review and run all unit tests. Fix any discovered bugs | Improved version of the web application | The QA tester will run all developed unit tests and work closely with programmer to fix any discovered bugs. | 4 days |
| Testing Phase | Create and run functional tests/Acceptance Test | Final version of the web application | QA tester will run all the functional tests ensuring the quality of the application. PM will coordinate with the customer to perform several of this test as an acceptance test. | 3 days |
| Deployment | Deploy the web application into production environment | Fully deployed web application | Deploy the final, tested, and accepted version of the web application to the cloud using Azure services | 2 days |
| Maintenance | Create a maintenance plan | (Optional) Maintenance service contract | Discuss maintenance options with the customer and offer our maintenance plan as an option | 1 day |
|  |  |  |  |  |