EAST TENNESSEE STATE UNIVERSITY

CSCI 4250 – Software Engineering I

Problem Identification and Definition (PID) Template

Overview

The PID is a high-level document that details what the customer wants in their product and how we are going to organize to create a base level of the software.

BucHunt

1. **Contact Information of Project Sponsor-**
   - Customer: Mr. Kinser, kinsersretired@gmail.com
2. **Business Objectives—**

   There are a number of business objectives for this product. First and foremost is to provide a more real-life development effort for students in this course. One which can be expanded on by subsequent semesters of students thus exposing them to having to work with an existing code base.

   Secondly, this product will provide ETSU with a more modern way to familiarize new students with the campus.

   Lastly, later renditions of this product have the potential to be a revenue generating solution for ETSU by allowing the university to white-label it for other organizations to conduct treasure hunts outside of the campus.

3. **Vision Statement—** The ultimate solution will be a browser-based white-labeled application that can be customized easily and supports a variety of revenue models. It will be developed using a tiered approach. Each tier will be developed by students as part of their 4250 course work. Each tier deployed will have an increasingly more customizable way to create a list of tasks that need to be performed as part of "Scavenger Hunt". This document is targeted at the first tier, or Iteration 1.

   In iteration 1, players will be given a URL and an access code by which they can access a scavenger hunt. The hunt will be a simple, fixed list of popular locations on the ETSU campus to visit. At each location, there will be a QR Code that they must scan to obtain a longitude and latitude point. Each player's progress is tracked on the server and displayed to the respective players. Once a player has completed all the tasks, they are sent a congratulatory email..

4. **Scope**

   The scope in iteration 1_1 builds on the previous implementation of the last semester. Thus, some of these items are already implemented but you must validate the implementation and expand on it. The scope for your iteration includes the following:
   - The list of tasks and player data must be persisted on a server.
   - There is only one hunt enabled for this iteration and it is always “active”.
   - Integration of QR Code scanning within the player interface is optional but desirable.
   - Integration of a graphical map within the player interface to show location of the tasks.
   - Support multiple players at the same time, each with unique access codes
   - Support multiple users using the same player access code
   - Indicate to players their current progress in the hunt
   - Ability to edit their username which defaults to a random number
   - Support admin functions

5. **User roles**—name and briefly define/describe the user roles.
   - Player: an individual (or team) who is playing / participating in the scavenger hunt
   - Admin: person(s) responsible for updating the server side information for the hunt and players. This person(s) will be responsible for sending any texts or emails to players that are not automated by the system.
   - Player interface: the browser based application the Player interacts with.

6. **User functionalities**—list the product’s major user functionalities (functional requirements).:

Players…
- shall be identified by their email **and** text enabled phone number.
- shall be assigned a human consumable and human friendly access code.
- shall be sent an access codes as a text message (“Thank you for Playing BucHunt: Your access code is <insert here> “)
- shall be sent the URL for the game via email at the same time as their access code is sent to their phone. (“Thank you for Playing BucHunt. The current hunt is accessed here <insert url> and your access code has been sent to your phone. Good Luck and have fun!”)
- shall be required to enter the game URL into a browser and enter a valid access code in order to begin playing.
- shall be presented with a fixed list of task labels upon entering a valid access code.
- shall have their game status automatically stored on the server.
- shall have the ability to manually enter the decoded QR Code from within the client interface to complete a task.
- shall have a username associated with them (defaults to a random number).

Game…
- shall store any additional information about each player on the server.
- shall support a single game and it is considered active at all times.
- shall associate multiple tasks with the game (including task label, GPS location, QR Code image, QR Code decoded value)

- shall track each Player's progress in completing the tasks for the game
- shall display to each Player, each task label in the list along with a completion indicator.
- shall display an error message to the user if they enter an invalid decoded QR Code
- shall update the server and display to indicate a task is complete when the Player enters a valid decoded QR Code
- shall store a timestamp in association with any completed task
- shall send a Congratulations message to the client interface and in an email when the Player successfully completes all tasks defined in the game.
- shall send each Player an email with their results and rank when the game is ended.
- shall refresh the client interface periodically to ensure the Player is seeing their actual progress and statistics.
- shall NOT use cookies or persist any data on the client devices.
- shall have the ability to edit their display name which defaults to a random number.

Nice to have functionality for Iteration 1 (desired but not critical):

- Players may have the ability to scan a QR Code from within the client interface which will complete a task; scanning an invalid QR Code will result in an error message to the Player.

Non-functional Requirements:

- The solution shall be deployed in a way that future classes can maintain
- The code base and associated documentation shall be stored in a way that future classes can maintain and extend
- Access Codes shall be unique to each player.
- QR Codes shall be unique to each task
- Task labels shall be unique to each task
-

7. Any special issues or considerations:

- Must be a client-server architecture.
- All data must be stored on the server (no cookies).
- Client interface is browser based (vendor neutral).
- Client interface is internet connected device neutral (e.g. works on phones, tablets, laptops, etc.)
- QR Codes must be unique within a single game.

Risks:

- Integration of QR scanners to the application may not be possible without the purchase of a third party application and subsequent integration with the main product.
    - Mitigation: since this is a "nice to have" requirement for Iteration 1, it will be tackled after all required functionality is functioning.
- Sending text messages from a server side application may not be possible without the purchase of a third party application and subsequent integration with the main product.
    - Mitigation: manually send all text messages required in Iteration 1 (this is just the access codes).

-
    -

8. **Glossary of Terms:**
    - **Web application** - Application software that runs on a web server.
    - **Client server model** - User (client) is provided services through an off-site server.
    - **QR code** - A machine-readable code consisting of an array of black and white squares, typically used for storing URLs or other information for reading by the camera on a smartphone.
    - **Client interface** - The mechanism by which a user interacts with an application, service or system.
    - **Browser based application** - An application that runs within a Web browser.
    - **Use case** - A specific situation in which a product or service could potentially be used. Used to show the flow of data through the program.