Treasure Hunt App and Engine

FUNCTIONAL SPECIFICATION

Laszlo Malina
C00157639

Table of Contents

Introduction

This document specifies the treasure hunt app and engine project original vision, the use cases detailed use cases and FURPS.

Vision

Project Title:

Treasure Hunt App and Engine

Project Description:

The project is going to have two components, a website and a mobile application.

Website:

The website will be used to create treasure hunts. These treasure hunts will consists of creating a specific path between the start and the destination. In order to create the treasure hunts the user will have to sign up and log on to the site. Then they will follow a few steps to create the treasure hunt.

Mobile Application:

The mobile application will allow users to download the treasure hunts which are closest to their location from the website. They will then follow the treasure hunt path. At each location a question will be asked that is relevant to the current location. The user must answer the question correctly in order to proceed to the next location. These steps will repeat several times. At the end of the treasure hunt the user will win a small prize.

Outcomes:

The outcomes of this project will be a website and a mobile application. The users will be able to create treasure hunts on the website and download them to their mobile phones. When they are finished they will be able to like or dislike and rate the activity. The aim of this project is to get users to upload their treasure hunts and also encourage them to go out and get to know their city.

Functionality:

Website:

The site will have the following functionalities. There will be an option with signup/login for accessing the content of the website. The next option will be to create treasure hunts, which will have the following details: Create path for treasure hunt, choose each stop up to the destination, and set question(s) for each stop. There will be an option to like/dislike the activity. And when the user is finished they can sign out.

Mobile App:

The mobile app will present a screen to find treasure hunts in the area. The chosen path of the treasure hunt will be downloaded to the phone. The mobile app will display a question at each stop until the destination. But the user must give the correct answer to proceed. At the destination the user will close the app or choose another treasure hunt.

Technology:

The target platform of the mobile app is going to be Android and will be available for download. I will use Android Studio which uses Java, to create the app The website will be hosted on blacknight. I will use Joomla(CMS) to create the frontend. I will also use HTML5 and JavaScript to create the pages individually. I will use PHP and MySQL to access , store and retrieve data from the database.

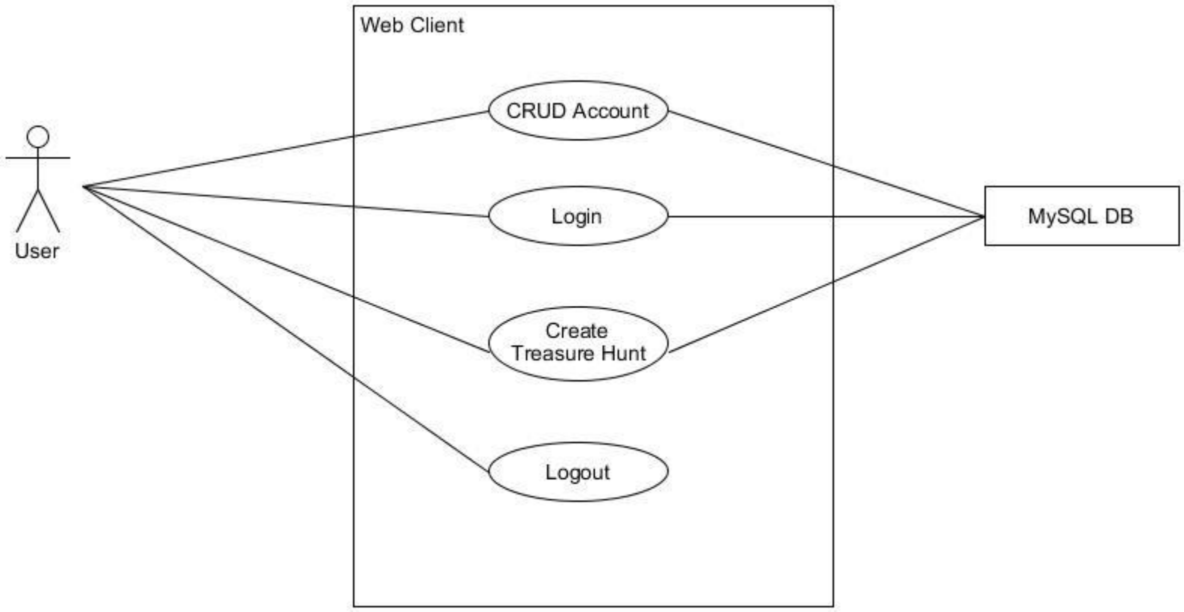
Potential Users:

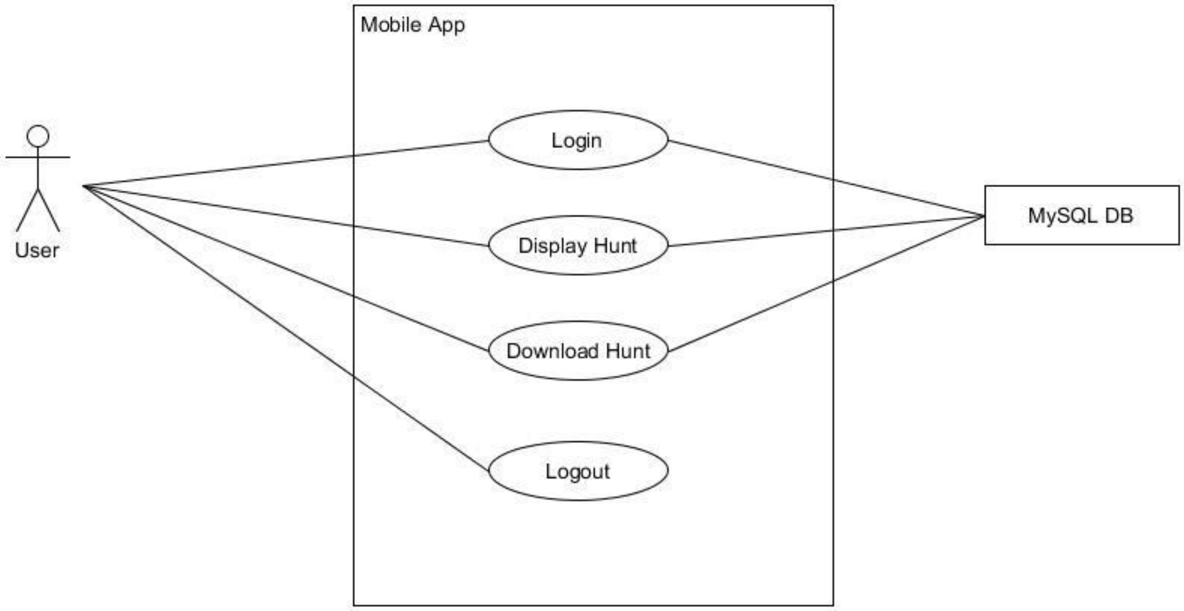
The project and application are for the general public, to make a large community who create and have fun using the treasure hunt app and engine. The app and site will be simple to use so the regular user can navigate just as easy on the site and the app as a more experienced user.

Data and Content:

The core of this project is to save treasure hunt locations and GPS coordinates to the database and then retrieve them later in order to use them. To make the site and the app connect to the database and send information to each other. The phone app also will determine what the users location is on a map.

Use Case Diagrams





Brief Use Cases

Web Client

CRUD account

Actors: User, MySQL DB

Description: This use case begins when the user wishes to create an account on the website. They will click on the create account option. The user must enter their information such as name ,chosen username , password and email address into the form provided and upon completing than submit it. Then this use case will finish.

Login

Actors: User, MySQL DB

Description: This use case begins when the user wishes to login to their account. They will click on the login option. The system will request the users login details such as username and password. The system will validate these details with the stored records within the system to verify that this is a valid account or not. The result of this action are returned to the user, with the system successfully approving the login. The use case then concludes.

Create hunt

Actors: User, MySQL DB

Description: This use case begins when the user chooses the create treasure hunt option after they are logged on. The user needs to choose a valid name for their treasure hunt. Then they will choose the route for the treasure hunt. For each chosen location in the treasure hunt, they will be able to create a question and answer to the next location. When they are finished , the system sends all the information to the database and this use case will conclude.

Logout

Actors: User

Description: When the user finished using the services of the web client they can choose to log out and when they do the use case will conclude.

Mobile App

Login

Actors: User, MySQL DB
Description: After the start up and information screen this use case begins. The user is presented with the login screen where they can enter their username and password. If the information they provided is matching the record in the database then they will be redirected to the next screen and this use case will conclude.

Download Hunt

Actors: User, MySQL DB
Description: After a successful login this use case begins. The user will be presented with the download hunt option. When the user presses the button to list treasure hunt they will get a list of current treasure hunts. When the user chooses one of them they will be moved to next activity and this use case now concludes.

Display Hunt

Actors: User, MySQL DB
Description: After the user chosen a treasure hunt this use case will begin. The user will be presented with the map with the first location of the treasure hunt and a text field to type their answer. When they click the marker on the map, they will then see the clue and the hint if there is any provided. The user now can provide an answer to the clue and if they are correct When there is now more clues to solve this use case then concludes.

Logout

Actors: User
Description: When the user finished using the services of the application they can choose to log out and when they do the use case will conclude.

Detailed Use Cases

Web client

CRUD account

Actors: User, MySQL DB

Typical course of events:

1. The user must enter their name, chosen username, password, validate password, email address validate email address on the form supplied
2. The information that is entered by the user has been validated
3. The user account now has been created and saved in the database
4. The use case then concludes

Alternatives:

3A. The username already exists.
The user must go back to step 2.
3B. The passwords do not match.
The user must go back to step 2.
3C. The email addresses do not match.
The user must go back to step 2.

Login

Actors: User, MySQL DB
Preconditions: The user already has an account

Typical course of events:

1. The user will enter their username and password on the login form
2. The information entered by the user now is validated, compared to the database data
3. The information is correct, the user is logged into their account
4. The use case then concludes

Alternatives:

2A. The username supplied by the user is invalid.
The user must go back to step 1.
2B. The password supplied by the user is invalid.

The user must go back to step 1.

Create Hunt

Actors: User, MySQL DB

Preconditions: The user is already logged in. No connection issues to the server and the internet and the database.

Typical course of events:

1. The user enters the desired name for their treasure hunt
2. The user now can specify a treasure hunt route on the map
3. For each location in the route they will enter a question and a correct answer
4. The system saves the locations with question and answers to the database
5. The use case then concludes

Alternatives:

1B. The name of the treasure hunt is already taken.
The user must return to step 1.

Logout

Actors: User

Preconditions: The user is already logged in

Typical course of events:

1. The user select the logout option.
2. Then the user is logged out.
3. The use case then concludes

Mobile App

Login

Actors: User, MySQL DB
Preconditions: The user already has an account created and connected to the internet.

Typical course of events:

1. The user will enter their username and password on the login form
2. The information entered by the user now is validated, compared to the database data
3. The information is correct, the user is logged into their account
4. The use case then concludes

Alternatives:

2A.The username supplied by the user is invalid.
The user must go back to step 1.
2B.The password supplied by the user is invalid.

Download Hunt

Actors: User, MySQL DB
Preconditions: The user is already logged in and connected to the internet.

Typical course of events:

1. The user chooses the download treasure hunt button
2. The user will be moved to the next activity
3. The use case then concludes

Display Hunt

Actors: User, MySQL DB
Preconditions: The user is already logged in and connected to the internet.

Typical course of events:

1. The user is presented with the map and the first clue.

2. The user knows the answer to the clue and enters it.
3. The answer is validated by clicking the button, compared to the database data.
4. Next clue is loaded and user must return to step 3.
5. Step 3 and 4 continues until there are no more clues to solve.
6. The use case then concludes

Alternatives:

3A. The user does not enter any answer and clicks button.
The user must go back to step 2.
3B. The user enter the incorrect answer.
The user must go back to step 2.

Logout

Actors: User
Preconditions: The user is already logged in

Typical course of events:

1. The user select the logout option.
2. Then the user is logged out.
3. The use case then concludes

FURPS +

Functionality

- The web client system stores the information of the users and treasure hunts.
- The mobile App system uses the stored information of the users and treasure hunts.

Therefore the functionality of the mobile App and engine requires big amount of data handling and processing.

Usability

- Generate an account on the web client is essential to access the functionalities of the web client.
- To use the mobile App functionalities it is necessary to have an account created by the web client.

Reliability

- The web client must be available 99.9% of the time based on the network availability.
- The mobile app must be available at 99.9% in order to receive the data from the web client.

Performance

- The structure of the project is not housing a large number of users therefore the performance is not going to be an issue
- Still the web client and the mobile app must stay stable when the users access their accounts or treasure hunts.

.

Security

- A register/ login system is implemented on the web client and the mobile app to make it more secure.
- There is also a change password facility in case the user feels their given password is not that strong.