Murder Mystery Adventure Game

Amelia Handley (40326169)

Abstract

The aim of this assessment is to design, create and implement a web application using the python flask microframework. The website being designed will be a text adventure game based on a murder mystery story. This also means that the website will have to be an enjoyable user-experience as well as simple to follow to avoid any confusion. This report will review all the technologies researched for the development of a functional site as well as show the initial wireframes of the design and the features of the website.

1. Description of Site

The murder mystery website will follow a text-adventure game format. The standard approach for this is to have the user input text or select decisions to control a character or the environment around it (Rollings & Ernest, 2010). In this case, the user will follow a murder mystery story and, depending upon their responses, will determine how the rest of the story is played out. There will be a forum page for users to discuss their own game experience and share their own choices they made. This will also mean setting up user profiles to enable people to communicate on the forums about the game, as well as an additional feature allowing users to save their progress on the game.

2. Background Research

3.1. Python Flask

Flask is a microframework written in Python as it does not require libraries or tools to use (Ronacher, 2013). As the aim of the module is to develop a better understanding of the language and to demonstrate it by producing a server-side web, further reading of Python Flask was undertaken. Firstly, following the workbook provided for the module allowed for initial use and insight into the language. Then looking at applications already created by others - which were like the goal of the murder mystery website, in python flask allowed for a further understanding of the language and additionally practice using it (“adventures-in-text”, 2013).

3.2. Bootstrap

The website will be designed using Bootstrap templates for design (including typography), buttons and navigation. Bootstrap is a free and open-sourced CSS framework which simplifies the development of web pages. It adds a variety of choice such as colour, size, layout of the page and the font (Otto, 2019). Bootstrap will be suited for the development of this website as it is responsive and easy to use, making it ideal for the development of a web game.

3.3. APIs

An Application Program Interface (API) is an interface between a client and a server. When a client makes a request to the web server, it gets a response back in a specific way or as a defined action (Braunstein, 2018). A RESTful API is an application interfaces that uses HTTP requests to GET, POST and DELETE data for web services. Requests made by the user on the website will, in the context of this website, elicit a response from a HTML file (Fielding, 2014).

An advantage of using RESTful APIs is that they are reliable. To illustrate, if there is failure between the components or data the system should be able to resist. Additionally, APIs are scalable. This means that it can support large numbers of components and the interactions between them (Wells, 2019). Therefore, API design would be favourable for this website as users will perform actions such as contributing to forums, updating profiles and perhaps deleting or editing comments (Smyth, 2019).

3.4. Cookies

An HTTP cookie, or a browser cookie, is a small piece of data sent from a website and it is stored on the user's computer browser. They are reliable pieces of software that remember user information (or stateful information) whilst a user is browsing (Vamosi, 2014). For the murder mystery game, the cookies would be used to store information such the decisions made by the user for the game or could be used as authentication for user log ins - with the appropriate security measures.

3.5. Textadventures

For inspiration for the murder mystery website, there was exploration into similarly created games online. A popular website for online adventure games is textadventures ("textadventure", 2019). Textadventures is a website which allows users to play and create their own text adventure games and share them with other users. To play games on the site does not require the user to have an account. An account can be created to allow the user save progress on games they play, to interact with other users and the creators of the game. They can do this either by commenting on a game or writing in forums dedicated to specific discussions about the games. This website relates to the creation of the murder mystery as the aim of the game is to create a website which allows users to play a game. It also will be designed and created to enable users to communicate on their own performance of the game as well as save their own progress.

To evaluate how similarly created games performed, a top-rated mystery game on the website was reviewed. The "Victorian Detective", by the user peter123, is described as a detective game which navigates through the streets of London, with the primary investigation being a mugging ("Victorian Detective", 2019). The story is displayed in a text box with choices for the user to select below. Once selected, there are prompts as to whether you made the correct or incorrect "detective reasoning" decision and the consequences of the actions. The game design is basic and understandable and follows an engaging narrative. The story is all on one page which so when the user makes a decision, it continues beneath making the game easy to follow. The only notable issue would be how the user selects their choice. They are labelled using blue hyperlinks. However, other keywords in the story use the same format (i.e. are underlined in blue). This may be an area of confusion for users so it would be more appropriate to make the decisions stand out more. The design of this game would be ideal on the murder mystery website to ensure that the user is able to understand the narrative as well as being interactive- allowing the users to make their own decisions.

4. Features

4.1. Text-Based Game

The main feature of the website will be the game itself. A user will be able to access the website and play the game, with or without an account on the website. Based on the options the user has selected will determine their outcome and their reward. The game itself will be set in a casino, inspired from a Murder Mystery game plot already published online (Breen, 2011).

The plot follows the events after an Elvis impersonator is killed suspiciously, leaving everyone at the venue a suspect. The user will assume the role as a detective and try and make the correct decisions to solve the murder. The decisions the user makes, after reading a section of the story, will be able to be selected. To win the game the user must select the correct suspect.

4.2. User Profile

To be able to save their progress on the game, users will be required to make their own accounts. By collecting an email, a username (or a name they wish to be called in forums and comments sections) and a password for each user creating an account, will allow users to have their own unique login. By collecting an email- which is already a unique attribute - will ensure that user-emails are not duplicated by comparing them to already created user accounts. This is how the database will search for a user and authorise user access to the website, by searching the user's email.

4.3. User Forum and Interaction

By creating a user account creates an opportunity between interactions of users. Whether it being on a forum or discussion post on the site or in the comments section of the game. There would be opportunity to expand user interactions to include "likes" or "dislikes" on both the game and on other topics that have been posted.

5. Website Implementation

5.1. Navigation Tree

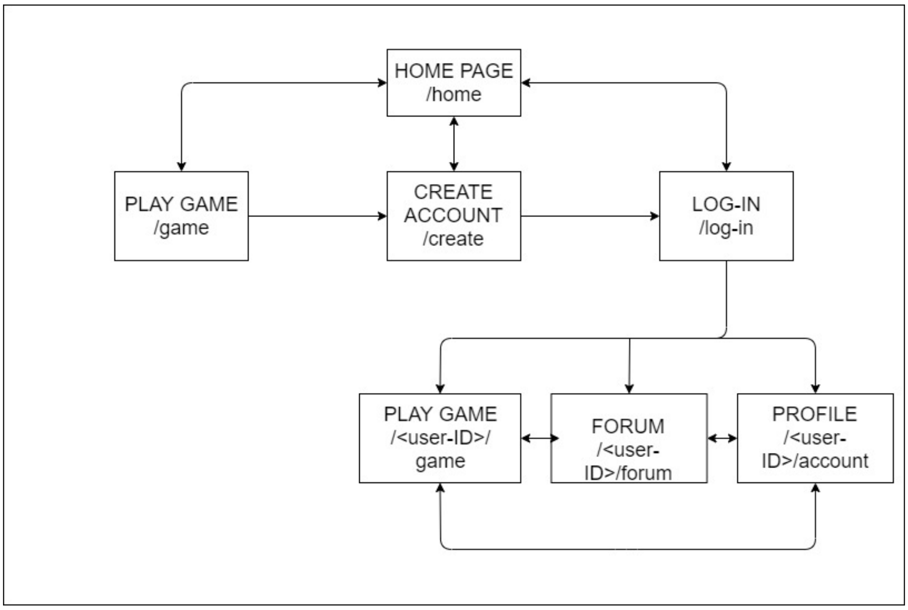

Image 1. Navigation Tree and URL-Hierarchy for the Murder Mystery Text Adventure Game

The site navigation will be simplistic (see Image 1). A user can play the game regardless of having an account. Therefore, from the home page their will be an option to either play the game, create an account or login. If the user decides to play the game without creating an account, they may decide they wish to save their progress whilst playing, which they can do by creating an account from the page. If the create an account option is selected the user can then create an account, then will be taken to the log in page. Once logged in the user will be able to access the game (which will allow them to save and view their progress), view their user profile or access the forum where they can comment.

5.2. URL Hierarchy

The URL hierarchy will follow the navigational hierarchy. To enable easy understanding and a consistent API structure requires a strategic naming system to be implemented (Saddey et al., 2019). In the case of this website, webpages such as the game are referred to as "/game" in the URL hierarchy (see Image 1). When logged in the user-ID (or name) will appear in the URL, which will be a unique attribute to each user, enabling the user to access the forum, profile and game pages. This makes the API intuitive and easy to use by a user.

5.3. Administration

Responsibility also falls on an administrator to maintain the website, for example, fixing bugs and enhancing the website. Website administration can involve managing user accounts, maintaining web servers, log analysis and most importantly web security. If a website is not secure it is liable to be attacked by hackers to gain access to user information with malicious intent ("Website Administration", 2019). Therefore, administration will be an important aspect of the website as it will allow the administrator control over the user profiles, for instance, if a user breaks terms and conditions or is impolite in the user forum, they can be removed. Additionally, they will be able to sort any issues with the website itself.

5.4. Database

To store this user data will require a database. SQLite3 is an easy to use database which already includes in built support in Python (Python Central, 2019). An advantage of using a database to store user information is that it can also be persisted. Meaning, that the user's information (including their user email, username and password) will be stored in a secure place-until the account deleted (Alsultanny, 2010). Additionally, once an account has been made, each user will have to enter a unique attribute - for instance the user email - which will be used as authentication to user the website. Therefore, the website will be able to save a user profile, authorise a log in (by searching database for the user's email) and save the progress of a game (when a user is logged into the website).

5.5. Security

To create the account the user will be asked to provide personal information there will need to be a security. Python Flask has a flask extension called Flask-Security which is used to add security mechanisms to applications ("Flask-Security", 2019). For instance, in the database it is not safe to store the user password in its true form. Instead a tool called password hashing is used to transform the it into a different string. This makes it almost impossible to go back to the original password. It does this by storing the initial password the user inputs to create the account as a hash. Then when the user logs back into the account, the database will compare the initial password with the one being used to access the account (Greenberg et al.,2016). This means that, when stored on the database, the users profile for the website will be almost impossible for a hacker or even the administrator to look at their data.

5.6. API Routes

To connect the users, which are not concerned with data storage, to the servers, which are not concerned about the user state, there are requests. Each of the requests contains all the necessary information to service the request. They use verbs such as GET, POST, DELETE to retrieve the web pages and servers. A GET request, for instance, would be used to collect the users information when the user is logging in. A POST request would be used when there is a new collection of data being entered - such as a new user account. A DELETE request might be used by the user to delete their account or delete a comment from the website (Wells, 2019).

6. User Interface

6.1. Website Design

The overall website design has been kept simple and minimalistic. This is to ensure that all the important elements are clearly labelled. For instance, the header will contain the key buttons to allow the user to switch between pages easily and will remain in the same place when moving between them. By keeping a similar format between the pages allows easy navigation and understanding for the user.

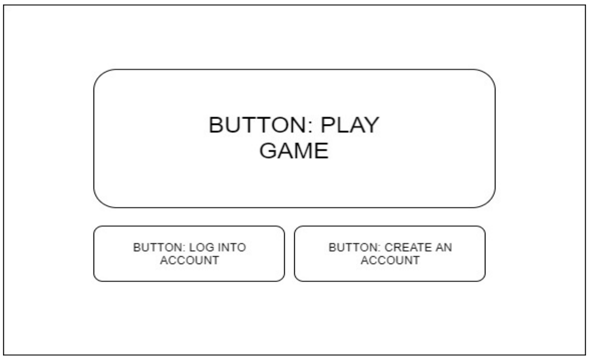

Image 2: Home Page

The home page gives the user three options (see Image 2). They can either play the game (without progress being saved), create an account or log into an account.

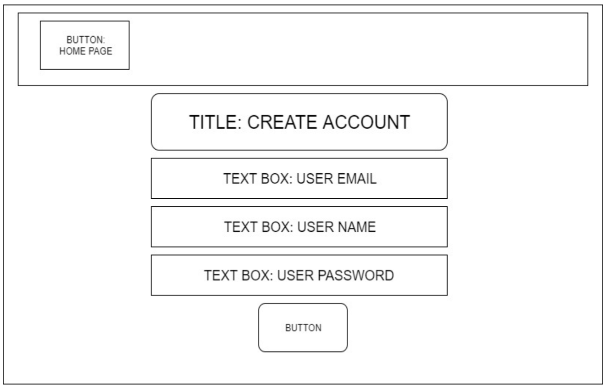

Image 3: Create an Account Page

The create an account page is simple and direct, meaning that it will ask the user exactly what it requires (Image 3). A valid email, a username and a user password will be required to make an account.

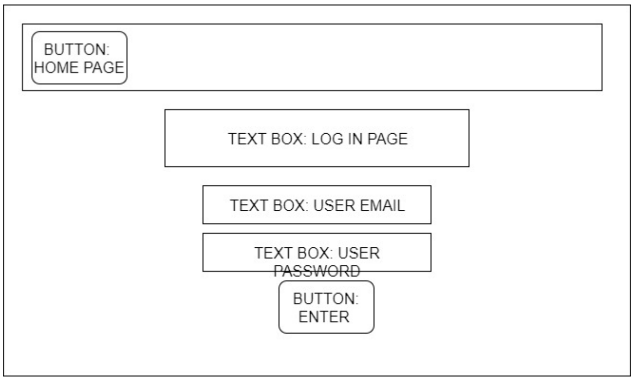

Image 4: Log-In Page

The login in page will ask for the user email and user password (Image 4). They can also return to the home-page if they accidentally selected the log-in page rather than the create an account page.

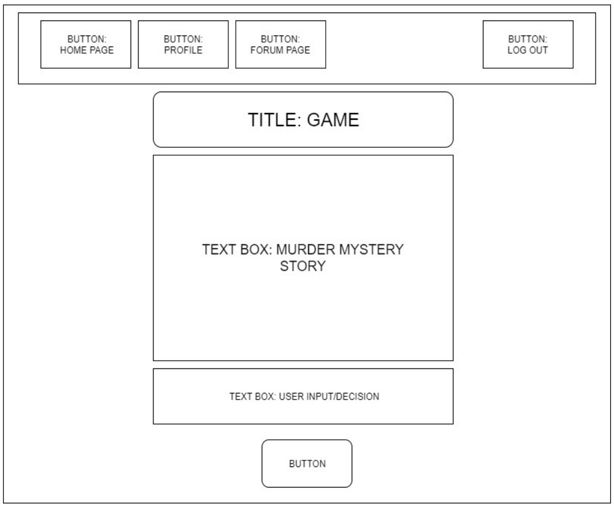

Image 5: Game Page

The user will then be directed to the Game Page (Image 5). The murder mystery game will contain the story in the text box. Based on the users input or decision will change the narrative.

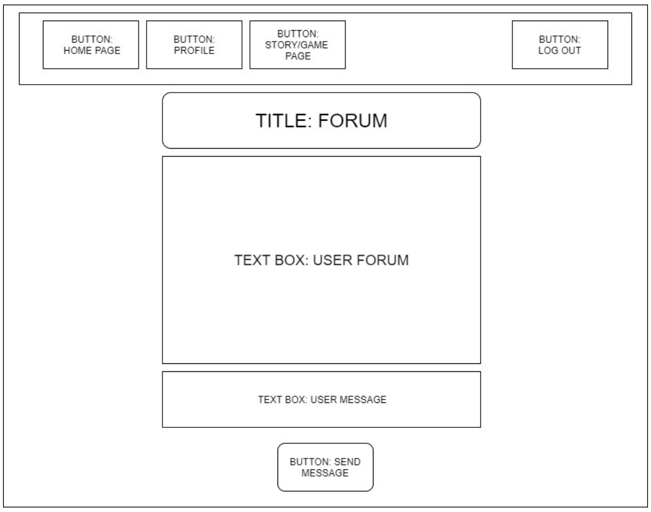

Image 6: User Forum Page

The user can also select to view their profile or the forum page as well as log out. The forum page will show different discussions occurring between users (Image 6). It will allow the user to comment or start their own discussion. They can also navigate between the Profile page and the Game page.

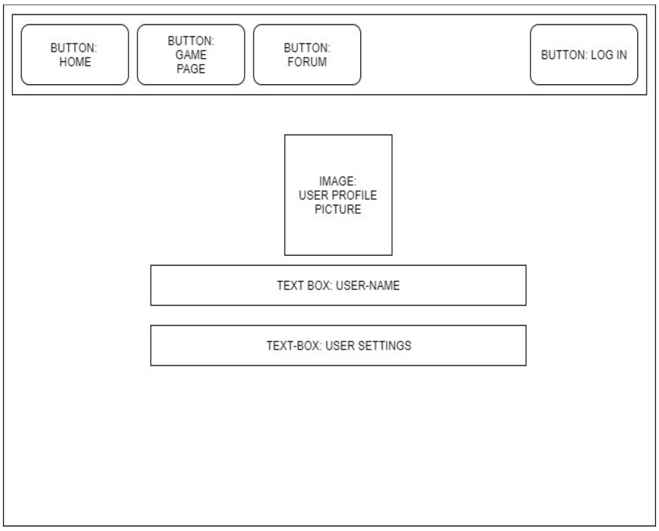

Image 7: User Profile Page

The user profile page will contain their username and any sensitive settings such as passwords (Image 7). Here users may select a profile picture they wish to appear as in the comments and forums

References

Alsultanny, Y. (2010). Journal of Database Marketing & Customer Strategy Management. Retrieved 27 October 2019, from https://link.springer.com/journal/41489

adventures-in-text. (2013). Retrieved 25 October 2019, from https://github.com/becdot/adventures-in-text

Braunstein, M. (2018). Health Informatics on FHIR: How HL7's New API is Transforming Healthcare. Retrieved 28 October 2019, from https://www.springer.com/gp/book/9783319934136

Breen, N. (2011). Free Murder Mystery Party Plan. Retrieved 28 October 2019, from http://nickbreen.ca/blog/free-murder-mystery-party-plan/

Fielding, R. (2014). RFC 7231 - Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. Retrieved 28 October 2019, from https://tools.ietf.org/html/rfc7231#section-4

Flask-Security. (2019). Retrieved 29 October 2019, from https://pythonhosted.org/Flask-Security/

Greenberg, A., Laslo, M., Barrett, B., Matsakis, L., & Newman, L. (2016). Hacker Lexicon: What Is Password Hashing?. Retrieved 29 October 2019, from https://www.wired.com/2016/06/hacker-lexicon-password-hashing/

Otto, M. (2019). Bootstrap. Retrieved 28 October 2019, from https://getbootstrap.com/

Python Central. (2019). Introduction to SQLite in Python. Retrieved 26 October 2019, from https://www.pythoncentral.io/introduction-to-sqlite-in-python/

Ronacher, A. (2013). Foreword — Flask Documentation (0.10). Retrieved 28 October 2019, from https://web.archive.org/web/20171117015927/http://flask.pocoo.org:80/docs/0.10/foreword

Rollings, A., & Ernest, A. (2010). Artificial Life and Puzzle Games. Retrieved 25 October 2019, from http://wps.prenhall.com/bp_gamedev_1/54/14053/3597646.cw/index.html

Saddey, R., Maunde, T., Yazzie, D., Escobar, J., Ishan, M., & Khor, Z. et al. (2019). REST API Naming Conventions and Best Practices. Retrieved 29 October 2019, from https://restfulapi.net/resource-naming/

Smyth, P. (2019). Creating Web APIs with Python and Flask. Retrieved 28 October 2019, from https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask#what-is-an-api

textadventure. (2019). Retrieved 26 October 2019, from http://textadventures.co.uk/

Website Administration. (2019). Retrieved 26 October 2019, from https://www.whitehatsec.com/glossary/content/website-administration

Wells, S. (2019). APIs and Web Services. Retrieved 28 October 2019, from https://www.dropbox.com/s/u6eszv5eo5dpxmq/L03_apis%2Bweb.services%2Brest.pdf?dl=1

Vamosi, R. (2014). Gmail cookie stolen via Google Spreadsheets. Retrieved 28 October 2019, from https://web.archive.org/web/20131209210402/http://news.cnet.com/8301-10789_3-9918582-57.html

Victorian Detective. (2019). Retrieved 26 October 2019, from http://textadventures.co.uk/games/view/w207ce4zekubenmwgss5pa/victorian-detective