Section A: Planning

Defining the Problem

My client, Mr. X, has recently graduated from university, and is hence taking a break to travel across the world. As a frequent traveler, he expressed a need to make his trips less cumbersome, both during the booking and travelling stages. In particular, he shared his frustration for manually and repeatedly conducting tasks such as sorting visas and other necessary documents to plan his travel. In our interview[1], he also mentioned inconveniences during his trips such as searching for nearby restaurants and forgetting to visit notable locations. In particular, he conveyed irritation towards constantly having to search for nearby locations and painstakingly browse through Google maps. Since he is an avid shopper, he also revealed how it can be easy to get carried away and avoid efficient budgeting. He further shared his love for writing daily reflections in a journal, but communicated concerns on the longevity of physical notebooks. Additionally, he mentioned that his "images are very randomly dumped in a file" [1] and that he would require a system that can effectively organize them. Moreover, he commented that he found current online applications ineffective as most of them are complex and require taking lengthy tutorials before one is able to easily utilize them. Furthermore, he asserted that current systems are unreliable and he feels unsafe adding sensitive information about him including private documents and images. He stated that "there is no guarantee that someone is illegally accessing my information" [1].

Word Count: 251

Rationale for the Proposed Solution

Post the discussion with the client, we have decided to go ahead with a local travel management system as it would solve all my client's problems with regards to travel. As my client mainly uses his laptop, I proposed a standalone local application as it would be more accessible than an Android or iOS application. I will use MySQL to manage all the data of my client including login information, images and documents, and Java to carry out all the algorithmic procedures. I will use Java Database Connectivity (JDBC) as it is the simplest application to connect Java with MySQL. Further, I will use Java Server Pages (JSP) and Servlet to launch the system onto the web. I have chosen these software especially because of their ability to run on multiple operating systems including macOS and Windows. To maintain high password security, I will implement the Argon2PasswordEncoder for

- high resistance against attacks
- multiple possible configurations.

I choose to work with the NetBeans IDE as it has a custom dashboard to show SQL queries and other activities. Since this is a large program and Java is object-oriented, it is also convenient to create modules and exploit the features of classes including inheritance, encapsulation and polymorphism. Moreover, on the backend, MySQL works well because of its

- reliability
- performance
- ability to effectively create a relational database for managing the links between my client's different activities

[1] Appendix 1.

Furthermore, as all these software have high compatibility, they are superior to using other software such as Python, which would need additional packages to connect to the system. This choice of software will ultimately satisfy the client's requirement for usability because of the ease to develop a complex system based on simple elements such as buttons and text fields.

Word Count: 265

Success Criteria

1. Account Creation and Updation
    a. Allow the user to create an account with their email and a secure password
    b. Send the user a verification email upon account creation
    c. Allow the user to change their password
2. User Authentication
    a. Allow the user to login to their account if the username and password entered is correct
3. Travel Expenses
    a. Allow the user to input their total travel budget and daily expenses
    b. Calculate their total travel expenses
    c. Show the user the percentage of their budget they have exhausted
4. Open Maps
    a. Allow the user to open and access Maps through the system
5. Travel Itinerary
    a. Allow the user to input their daily itinerary
    b. Allow the user to save and update their itinerary
6. Digital Journal
    a. Allow the user to add, delete and edit the journal daily
    b. Allow the user to mark each write-up with a specific date
    c. Allow the user to view the journal
7. Click an Image
    a. Allow the user to click an image using the system
8. Important Contact Information
    a. Allow the user to add important contact details
    b. Allow the user to mark certain contacts with high importance, some with moderate importance and others with low importance