**Software Development Plan Template**

**Description of Smart Traveler**

**Smart Traveler**

**12/3/2023**

1.0

**Presented To:**

Dr Mohamed Ramadan

**Submitted By:**

Hassan Baraka Zakir  
Omar Mohamed Elhassan

Monda Atef

Omar Mohamed Abozeid  
Josef Adel Adeeb

Kareem Ahmed Abdel Azeem

**REVISION HISTORY**

| **Date** | **Author** | **Distributed to** | **Version** | **Description** |
| --- | --- | --- | --- | --- |
| DD\ MM\YYYY | name | who | Matches title page | Brief description of change |

**TABLE OF CONTENTS**

1. **PRODUCT DESCRIPTION**
2. **Product Description :**

**Our application is all about for tourist that comes to luxor and don’t know how to spend their days so our application asks for the budget and the days he will spend and our application give him recommendation to some places by their owners who’s sharing their landmarks and restaurants and you can see the evaluation of theses places and their prices and there are comments bar to give a feedback**

1. **Client Description :** 
   1. **Tourist : coming for Luxor don’t know where to go so our application assist him to know where to go and how to spend their days by our application he just add his budget and his holiday days and our application give him a recommend**

* 1. **Owner of Landmarks : most of them want to share their location and their places so by our application he can just add his place and the price and all place field with evaluation from the clients**

1. **Goals :**

**Facilitate the tour of tourist so he can where to go and how to go and give him a fabulous tour and help the owners of landmarks to share their places**

**2. TEAM DESCRIPTION**

|  |  |
| --- | --- |
| Skills | Team Members |
| * Intermediate in documentation and SRS tasks * Know in interfaces and related functions of Flutter | Hassan Baraka |
| * Moderate in AI and machine Learning tasks | Monda Atef |
| * Professional in documentation UML diagrams and SRS * Intermediate in Database * Amateur in AI | Omar Mohamed Elhassan |
| * Know in UML diagrams | Kareem Ahmed Abdel Azeem |
| * Mobile Developer | Josef Adel Adeeb |
| * Mobile developer | Omar Mohamed Abozied |

No there is no SME it’s a simple project in a short Luxor range

What expertise is missing?

**3. SOFTWARE PROCESS MODEL DESCRIPTION**

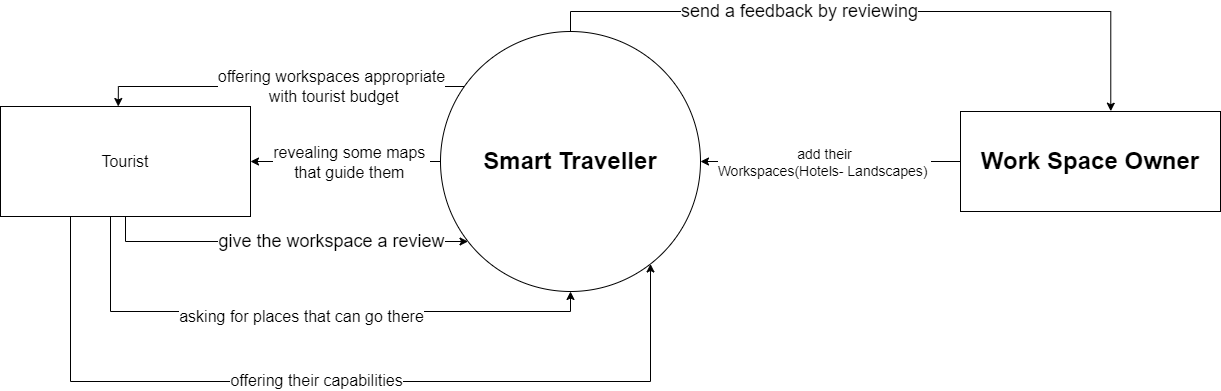
Agile, because it’s simple and affordable and can be delivered quickly

**4. PRODUCT DEFINITION**

Tourist : is the one who wants to know where to go with his capabilities (budget -spent days)

And reviewing the workspaces

Workspace Owners: they offering their places and their prices via our application by adding their locations

**Context Diagram**

**Personas**

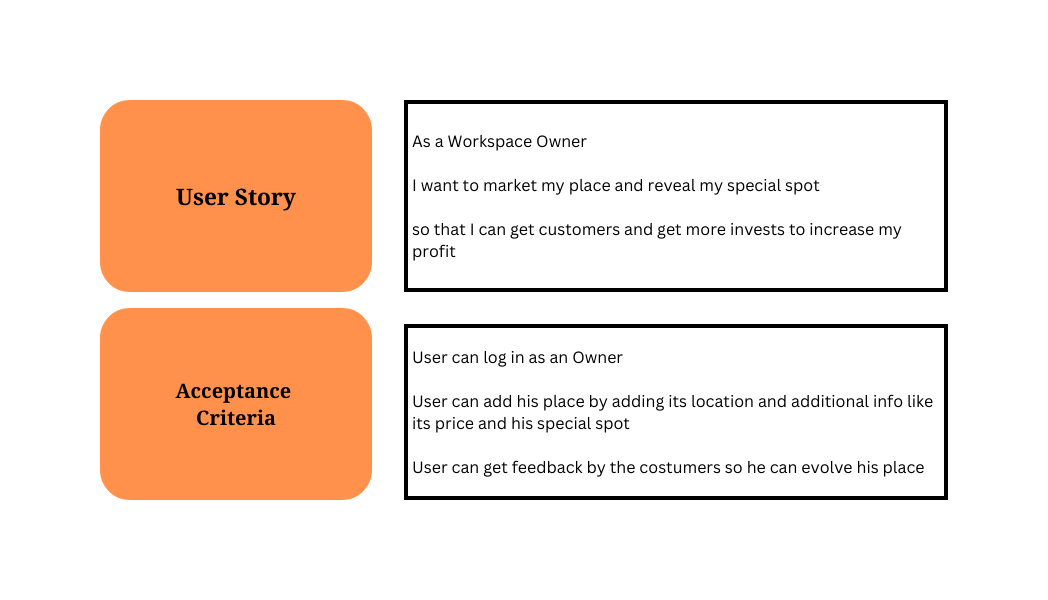
|  |  |  |  |
| --- | --- | --- | --- |
| User Persona | Who are they | What is their main goal | What is their main concern /barrier to achieve this goal |
| Tourist | Is the user that the whole application is made for him the application describes the services that it gives to him | He comes for getting a new experience and take a trip and hang out in a new country | Don’t know where to go and what to do and how to consume his budget and what are the important places to visit |
| Workspace Owner | The Skipper who’s offering his places for visiting and hosting with their location and prices | Gain Customers and a regular client for their places | Don’t know how to marketing his space |

**User Stories**

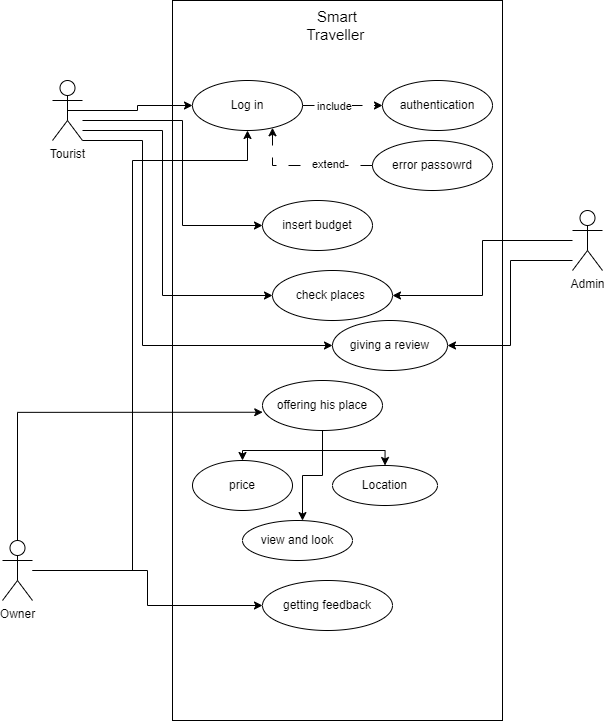
**Tourist Story :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | | **Priority** | **Estimate** |
| As a Tourist  I want a product that ease to me visiting and knowing how to get this and go there how to be thankful for my experience  So that I won’t need to inquiring people and get lost in a foreign country | | | |
| **Acceptance**  **Criteria** | User can login as a Tourist  User can add his information about where to go with respect of his budget  Offering places and their locations on a map and estimated days to spend  Give him review to give his opinion about the place (Hotel -Landscape) | | |

Owner Story :



**High Level Use Cases**

****

**Use Case Descriptions**

**Login Statement:**

1. **Unique name: Login**
2. **Participating actors: Tourist , Owner**
3. **Entry conditions: User has a registered account and has navigated to the login page.**
4. **Exit conditions: User is successfully authenticated and redirected to the homepage, or authentication fails and an error message is displayed.**
5. **Flow of events:**

**User enters their email and password.**

**System verifies that the email and password are valid and associated with an existing account.**

**If authentication succeeds, the system creates a session for the user and redirects them to the homepage.**

**If authentication fails, an error message is displayed and the user is prompted to try again.**

1. **Special requirements: The system should securely store user passwords and use encryption to protect them from unauthorized access. The login page should have proper input validation to prevent common attacks like SQL injection and cross-site scripting (XSS). Additionally, the system should implement measures to prevent brute force attacks, such as locking accounts after a certain number of failed login attempts.**

**Search a place Statement:**

1. **Unique name: Search Place**
2. **Participating actors: Tourist, System**
3. **Entry conditions: User has navigated to the search page and has entered a query for a specific place.**
4. **Exit conditions: System returns a list of matching places or indicates that no matching places were found.**
5. **Flow of events:**

**User enters a query for a specific place in the search bar.**

**System processes the query and retrieves a list of matching places from a database or API.**

**System displays the list of matching places to the user, including relevant details such as the name, address, and rating.**

**User selects a place from the list to view more details.**

**System displays additional information about the selected place, such as photos, reviews, and hours of operation.**

**User can choose to navigate to the selected place or save it for future reference.**

1. **Special requirements: The system should have a robust and accurate search algorithm to ensure that relevant results are returned for user queries. The search page should also have proper input validation to prevent common attacks like SQL injection and cross-site scripting (XSS). Additionally, the system should retrieve place information from a trusted source and ensure that user data is stored securely to protect privacy.**

**Insert Budget Statement:**

* 1. **Unique name: Insert Budget**
  2. **Participating actors: Tourist**
  3. **Entry conditions: User has opened the budget application and navigated to the budget creation page.**
  4. **Exit conditions: System saves the budget and returns the user to the main budget dashboard, or indicates that the budget could not be saved due to an error.**
  5. **Flow of events:**

**User selects the "Create Budget" option from the main menu.**

**System displays a form for the user to enter the details of the new budget, including the budget name, start and end dates, and budget amount.**

**User enters the budget details and submits the form.**

**System validates the budget details to ensure that they meet any requirements or constraints.**

**If validation succeeds, the system saves the budget and returns the user to the main budget dashboard, where the new budget is displayed.**

**If validation fails, the system displays an error message and prompts the user to correct any errors.**

* 1. **Special requirements: The budget application should have proper input validation to prevent common attacks like SQL injection and cross-site scripting (XSS). The system should also ensure that budgets are saved securely and that user data is protected. Additionally, the system may need to enforce budget constraints, such as limiting the number of budgets that a user can create or preventing overlapping budget periods.**

**Offer Place Statement:**

1. **Unique name: Offer Place**
2. **Participating actors: Owner, System**
3. **Entry conditions: User is logged in and has navigated to the "Offer Place" page.**
4. **Exit conditions: System saves the offer and displays a confirmation message to the user, or indicates that the offer could not be saved due to an error.**
5. **Flow of events:**

**User selects the "Offer Place" option from the main menu.**

**System displays a form for the user to enter the details of the place they want to offer, including the location, price, availability, and any special requirements or conditions.**

**User enters the place details and submits the form.**

**System validates the place details to ensure that they meet any requirements or constraints.**

**If validation succeeds, the system saves the place offer and displays a confirmation message to the user.**

**If validation fails, the system displays an error message and prompts the user to correct any errors.**

1. **Special requirements: The system should have proper input validation to prevent common attacks like SQL injection and cross-site scripting (XSS). The system should also ensure that place offers are saved securely and that user data is protected. Additionally, the system may need to enforce constraints on the number of offers that a user can make, or require additional information such as photos or reviews to verify the quality of the offered place. The system should also provide a way for users to edit or delete their offers, and notify them of any changes or updates to their offers**

**5. USER EXPERIENCE WIREFRAMES**

Initial prototype screens to validate initial understanding of the product.

**6. PROJECT ORGANIZATION**

Breakdown of major tasks and schedule

**Matrix of Responsibilities**

Defines the high level which team members are responsible for which tasks

**PERT / Gantt Chart**

First cut at schedule

**7. VALIDATION PLAN**

**Test Strategy**

What is the definition of done?

What does success look like?

**8. FEASIBILITY STUDY**

What are the known risks and how will they be handled?

**Risk Identification**

Description of the risks

**Risk Prioritization**

Prioritized list (biggest risk -> lowest risk)

**Risk Mitigation**

How will risk factors be addressed? By when?

What are you going to prototype?

**9. CONFIGURATION AND VERSION CONTROL**

Specify the process and attributes for version control for all project and product artifacts

**10. TOOLS**

Provide a list of tools required for the project and their use

**11. ARCHITECTURE**

List of hardware or other subsystems required for the product.