Name: Drashti Shah

Roll No: 1991044

SAP ID: 57498190044

**SRS FOR A ADVENTURE AGENCY MANAGEMENT SYSTEM**

**Table of Contents**

Table of Contents for a SRS Document

1. **Introduction**
   1. Purpose
   2. Document Conventions
   3. Intended Audience and Reading Suggestions
   4. Project Scope
   5. References
2. **Overall Description**
   1. Product Perspective
   2. Product Features
   3. Use Classes and Characteristics
   4. Operating Environment
   5. Design and Implementation Constraints
   6. Assumptions and Dependencies
3. **System Features** 
   1. Functional Requirements
4. **External Interface Requirements**
   1. User Interfaces
   2. Hardware Interfaces
   3. Software Interfaces
   4. Communication Interfaces
5. **Nonfunctional Requirements**
   1. Performance Requirements
   2. Safety Requirements
   3. Security Requirements
   4. Software Quality Attributes

**1.INTRODUCTION**

1.1 PURPOSE

This document gives detailed functional and nonfunctional requirements for the adventure agency management system. The purpose of this document is that the requirements mentioned in it should be utilized by software developer to implement the system.

1.2 DOCUMENT CONVENTIONS

* DB – Database
* ER – Entity Relationship

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

* This project is a prototype for the adventure agency management system.
* This project is useful for the adventure agency team and as well as to the customers.

1.4 PROJECT SCOPE

Adventure agency system will be applicable everywhere, where agency’s exists. It will be more efficient and easier way to have a record a system through which everyone can easily access it according to his rights as compare to traditional booking for adventure agency system which included a lot of paper work. Every agency will prefer the technical booking system instead of the traditional booking system as it contains many useful features and fastest methods for the transactions.

1.5 REFERENCES

* <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
* <https://docshare01.docshare.tips/files/7641/76416736.pdf>

**2. OVERALL DESCRIPTION**

2.1 PRODUCT PERSPECTIVE

* Agency Details:

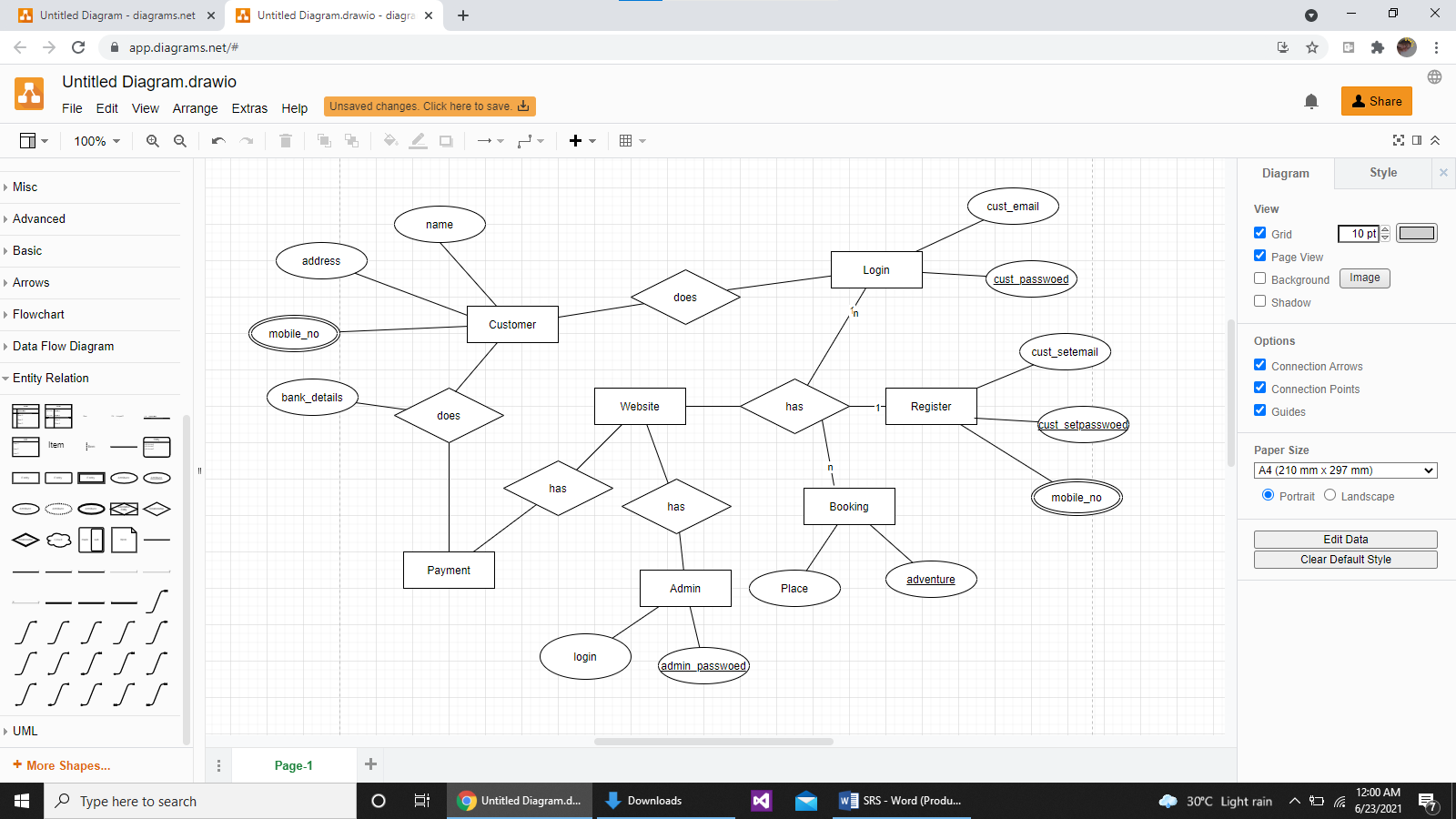
It includes details about the agency, its address, its employee’s

Customer description:

It includes customer code, name, address and phone number and other personal details. This information may be used for keeping the records of the customer for any emergency or for any other kind of information.

2.2 PRODUCT FEATURES

The major features of adventure agency management system as shown in below [entity–relationship model](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model) (ER model)



2.3 USER CLASS and CHARACTERISTICS

Users of the system should be able to retrieve information of their respective bookings of their future trips. The System will support two types of privileges i.e., giving information and booking. Customers will have access to only customer functions which are browsing.

The Customer should be able to do the following functions:

* Login/register
* Browse different adventures of different country
* Booking
* Payment
* Exit

2.4 Operating Environment

Operating environment for the Adventure agency management system is as listed below.

* client/server system
* Operating system: Windows.
* database: phpMyAdmin for MySQL over web XAMPP.
* platform: Python 3.9.2

2.5 DESIGN and IMPLEMENTATION CONSTRAINTS

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
4. Implement the database at least using a centralized database management system.

2.6 ASSUMPTION DEPENDENCIES

Following are the assumptions and dependencies which are related to this adventure agency project.

1) This project is a stand-alone project so it will not affect the system where it will be embedded.

2) This project is a desktop application project while the staff was addict of using traditional methods of data storage and retrieval so they will be trained a bit to jump to it.

3) This system will not depend on any other module.

4) It is will not affect the environment at all.

5) Adventure agencies will feel free to adopt it because it will not be so much expensive.

6) As this project contains valuable and new features so it will probably remove the previous adventure agency systems embedded in some agencies.

**3. SYSTEM FEATURES**

* DESCRIPTION AND PRIORITY

The Adventure agency system maintains information about all the customers, booking history regular customers, new registration, etc. This project does not have a high priority as the functions can be accessed whenever we wanted.

* **STIMULUS/RESPONSE SEQUENCES**
* Login or register
* Booking of adventures.
* Payment and cancellations.
* Functional Requirements

CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

* Some sites are client sites and others are server sites.
* All the data resides at the server sites.
* All applications execute at the client sites.

## EXTERNAL INTERFACE REQUIREMENTS

## USER INTERFACES

* + Front-end software: Python 3.9.2
  + Back-end software: MySQL

## HARDWARE INTERFACES

* Windows.

## SOFTWARE INTERFACES

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Database | To save the flight records, passengers record we have chosen MySQL database. |
| Python | To implement the project, we have chosen Python language is used in scientific and mathematical computing |

4.4 COMMUNICATION INTERFACES

This project supports all types of OS supporting python.

## NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of adventure agency database are as listed below.

NORMALIZATION:

* The basic objective of normalization is to reduce redundancy which means that information is to be stored only once.
* Storing information several times leads to wastage of storage space and increase in the total size of the data stored.
* If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table.
* Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem.
* These can be eliminated by normalizing a database.
* Normalization is the process of breaking down a table into smaller tables.
* So that each table deals with a single theme.
* There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes.
* It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

* 1. SOFTWARE QUALITY ATTRIBUTES
* AVAILABILITY: Any information about the account should be quickly available from any computer to the authorized user. The previously visited customers data must not be cleared.
* CORRECTNESS: Application should be correct in terms of its functionality, calculations used internally and navigation should be correct.
* MAINTAINABILITY: The application should be maintainable in such a manner that if any new requirement occurs then it should be easily incorporated in an individual’s module.
* USABILITY: This can be measured in terms of ease of use. Application should be user friendly. Should be easy to learn. Navigation should be simple.