Explore Egypt

Software Design Specification (SDS)
Version: 1.1

CM Identifier: G12_SE02_v1.1

Revision History

Sl. No.	Prepared/ Modified by	E-mail	Version	Date	Approved by	Descriptions/ Remarks
1.	Chahira Hamza	Chahira.hamza@live.com	1.0	04-09-2018		Design Pattern MVC
2.	Shrouk Ismail		1.2			Database Model
3.	Nooran Helmy	Nooran.helmi@hotmail.com	1.1	04-10-2018		Introduction System Architecture Class Diagrams Interaction Diagrams Design Pattern Factory Method
4	Chahira Hamza		1.3			Interaction Diagrams Deployment, Component Diagrams Traceability Matrix

Distribution list

Name	E-mail	Notes
TA: Eng. Ali Elseddeek	alielseddeek@gmail.com	
TA: End. Dina Elreedy	dinaelreedy@gmail.com	

Table of Contents

Index of Tables

List of Illustrations

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to set guidelines for the implementation phase. It provides an overview of the system architecture and the design patterns used. In addition to class diagrams, interaction diagrams, data model, system deployment diagram and Traceability matrix for illustration

1.2 Scope

Explore Egypt is a web application that are intended to encourage tourism in Egypt. It facilitates to the tourist the sites s/he can visit in every governorate in Egypt, gives information about the sites and their history. It also targets domestic tourism as it reveals the hidden places in Egypt.

1.3 Table of Acronyms and Definitions

Term	1.3.1 Definition
SDS	Software Design Specifications
GUI	Graphical User Interface
DB	Database
MVC	Model View Controller
CD	Class Diagram

1.4 References

Diagram Maker | Online Diagram Software. (n.d.). Retrieved April 10, 2018, from https://creately.com/

Fakhroutdinov, K. (n.d.). UML Diagrams Examples. Retrieved April 10, 2018, from https://www.uml-   diagrams.org/index-examples.html

1.5 Overview of Document

1.   The document describes the system architecture and the main components of the project. Then it describes the design patterns chosen and the reason for choosing them. Illustrates the dependencies between classes by class diagrams, the interaction between the user and the system by interaction diagrams, the classes and their data found in the database and the relationship between them by data model. In addition to deployment diagram and Traceability matrix.

2. System Architecture

The System architecture provides an overview of the system's major components and architecture.
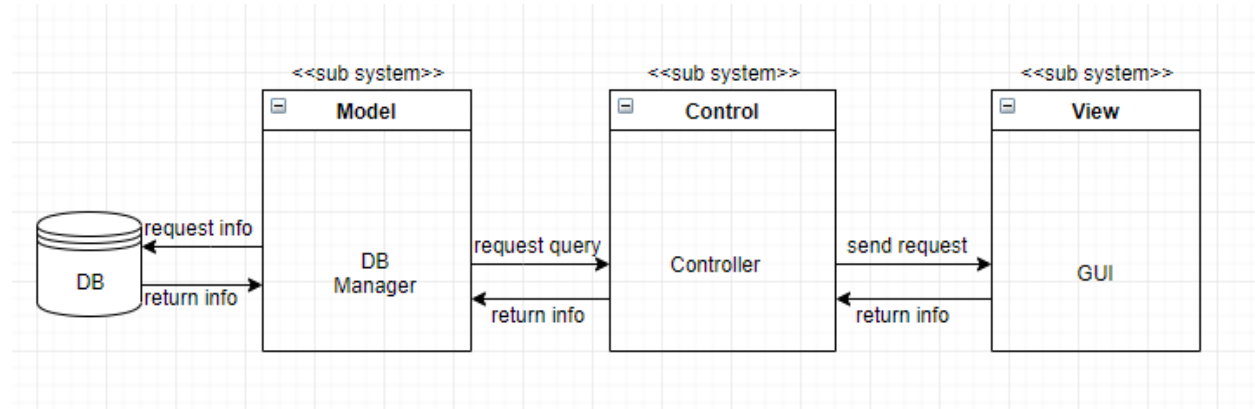

Illustration 1: System Architecture

The architecture of the project is following the MVC model, it is composed of three layers. The first layer is the GUI through which the user navigate through the website and use the functionalities. The second layer is the controller which controls which functionality each user should have and request a query from the DB manager depending on the functionality the user wants. The third layer is the Model and the database, the model gets the information which is stored in the database and passes it to the controller which passes it to the view so that the user can see it.

3. Design Models

3.1 Design Patterns Description

3.1.1 Design Pattern I : MVC

We have used the architectural MVC model in order to separate the 3 modules (Model, View, Control) from each other, so that each module can be independent. This design pattern facilitates the division between the team members in terms of tasks. The GUI representing the View module aggregates all the html files in use in the project. For each input from the user, a servlet is created which is a mini Control module that controls only this specific input. Finally a Model class that contains all the needed operations in the project and which handles the connection with the database.

3.1.2 Design Pattern II: Factory Method

We have used the factory method in order to create different user types. All users have the basic data in common while some types of users have additional data, so the factory method is the suitable design pattern for creating them.
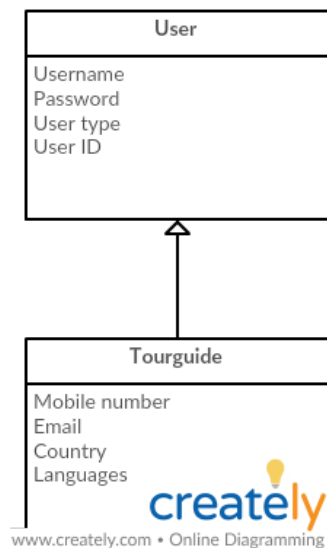
3.2 Class Diagrams

Illustration 2: User
Inheritance CD
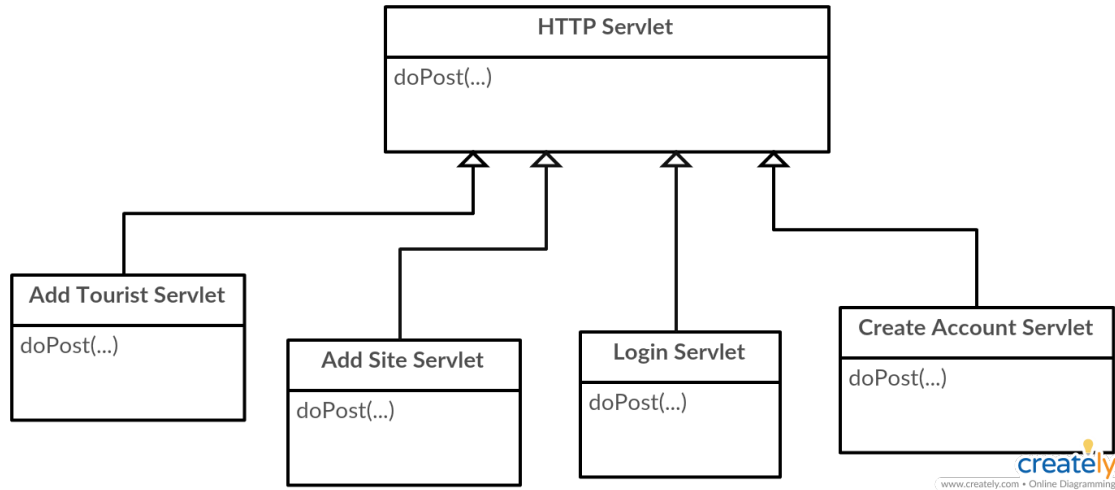
Illustration 3: Servlets Control CD
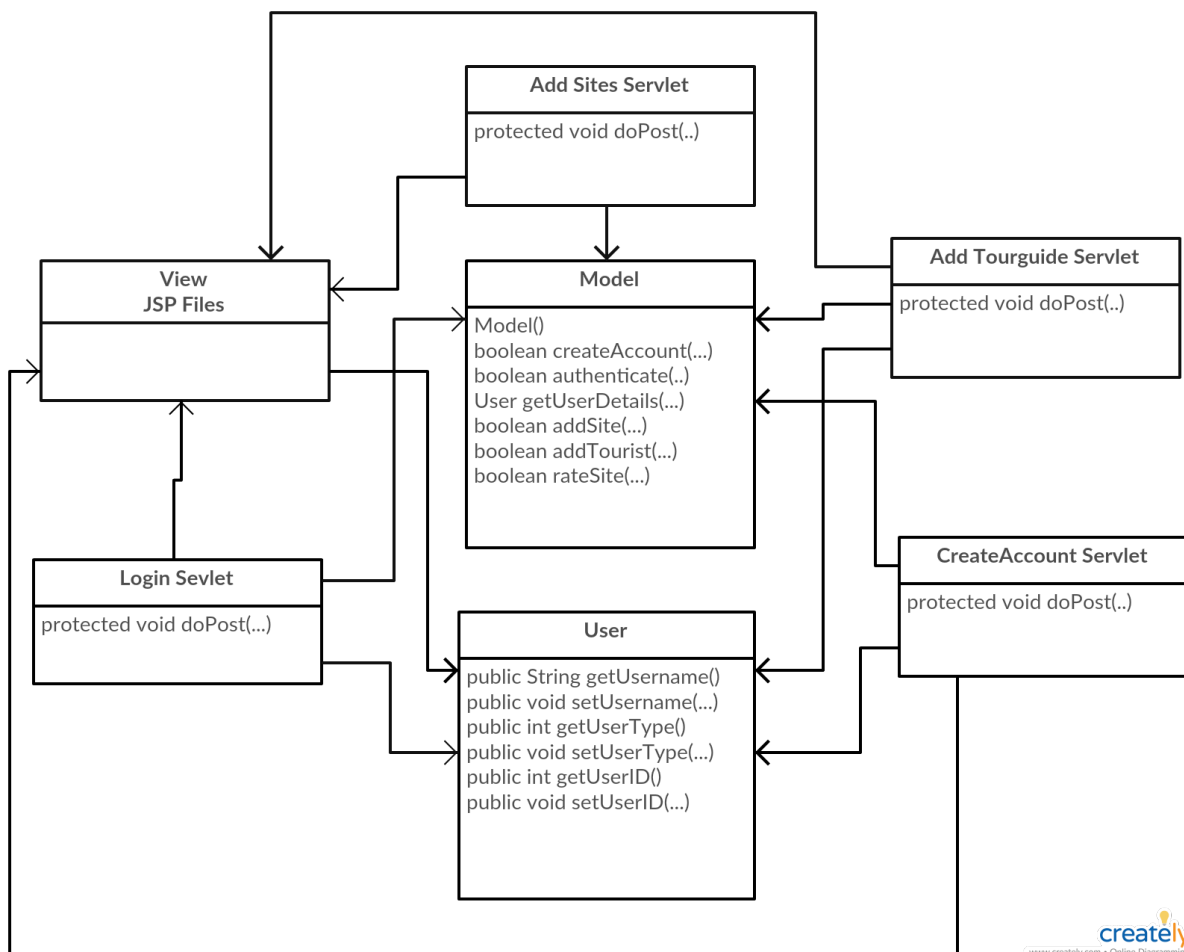
Illustration 4: MVC CD

3.3 Interaction Diagrams
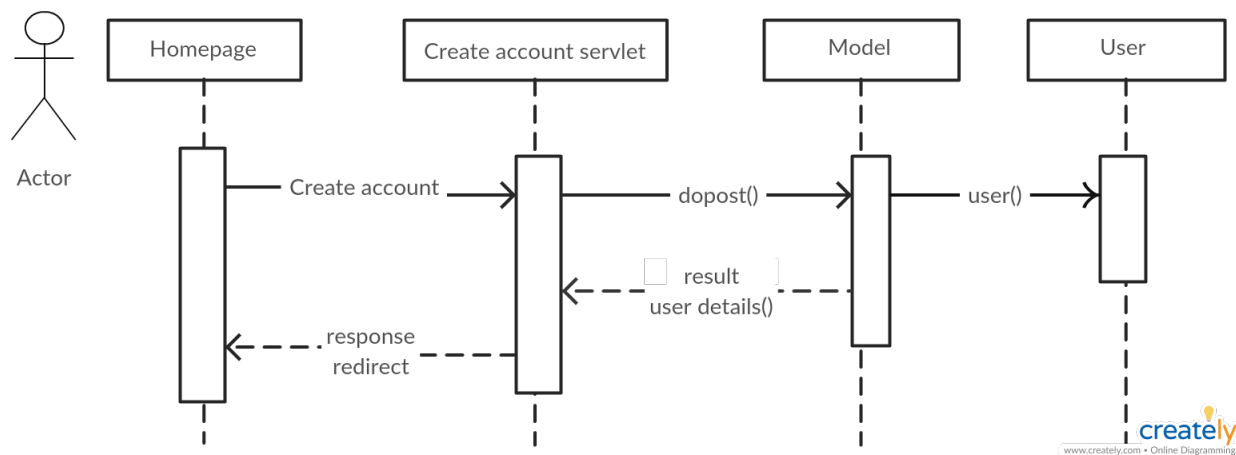
Illustration 5: UC 1.0
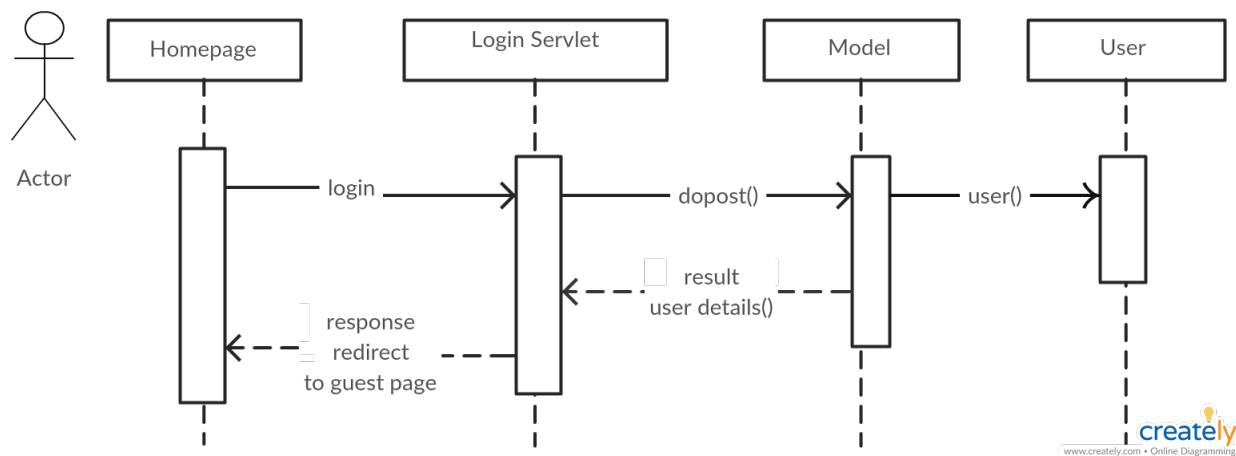
Illustration 6: UC 1.1
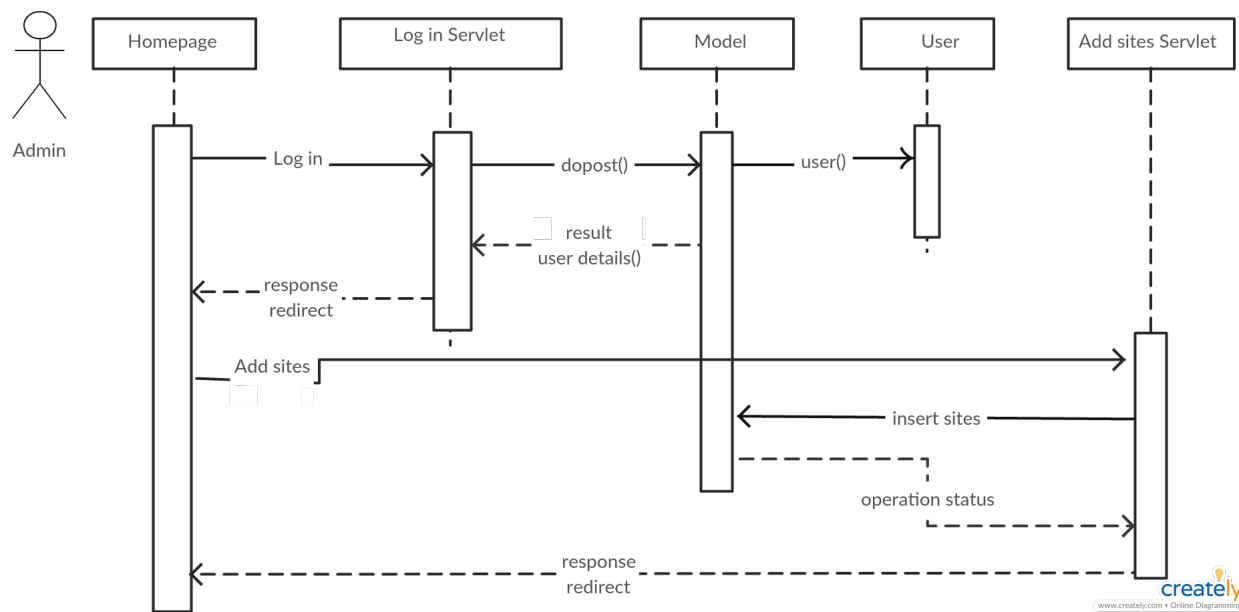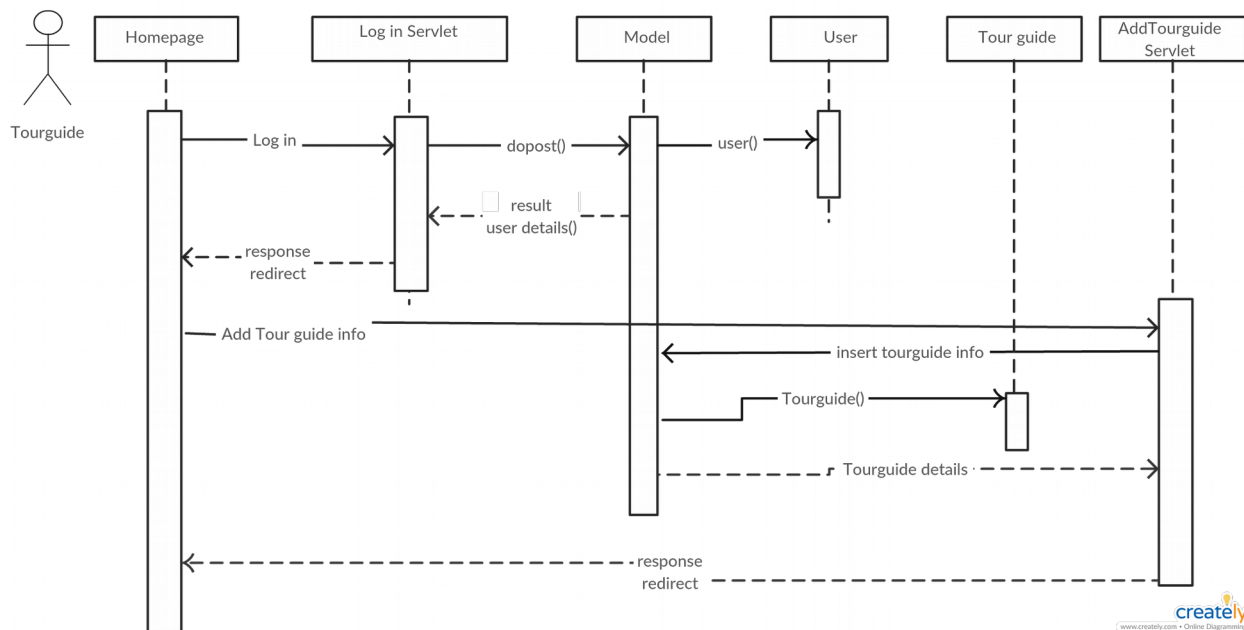
Illustration 7: UC 1.2

Illustration 8: UC 1.9
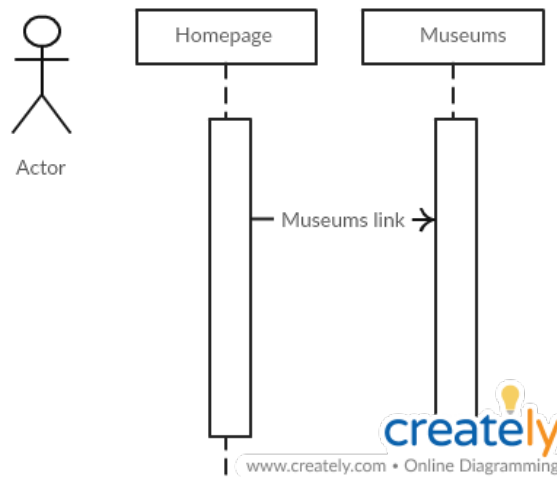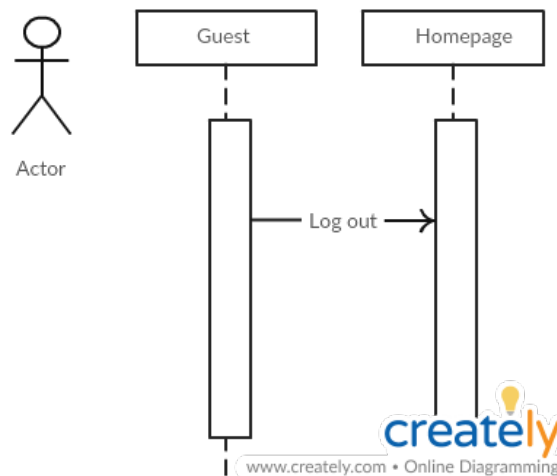
Illustration 9: UC 1.11

Illustration 10: UC 1.12

3.4   State Chart Diagrams (if needed)

Not needed

3.5   Activity Diagrams (if needed)

Not needed

3.6   Component Diagrams (if needed)

Not needed

4.   Data Models
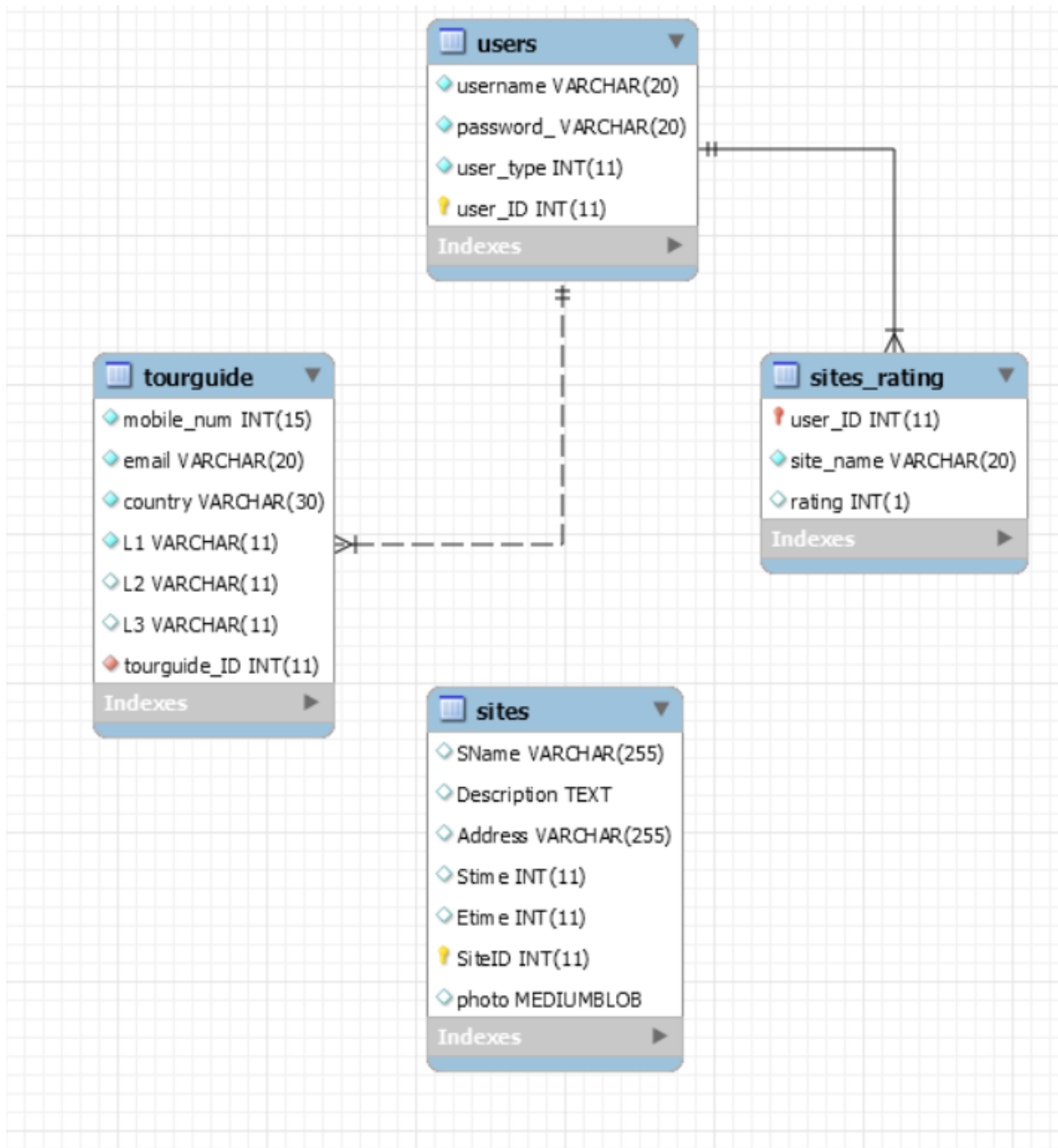
Illustration 11: Database Models

5.   System Deployment
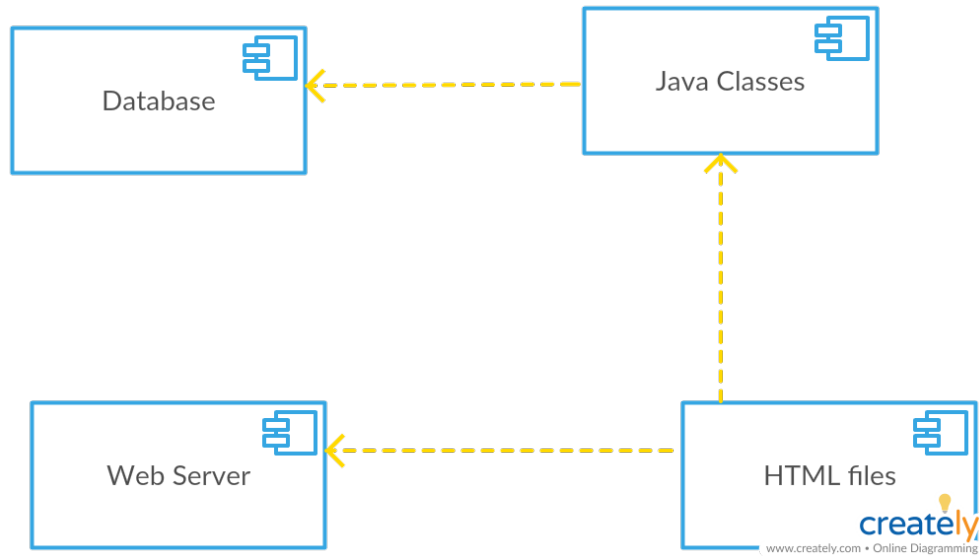
Illustration 12: Component Diagram
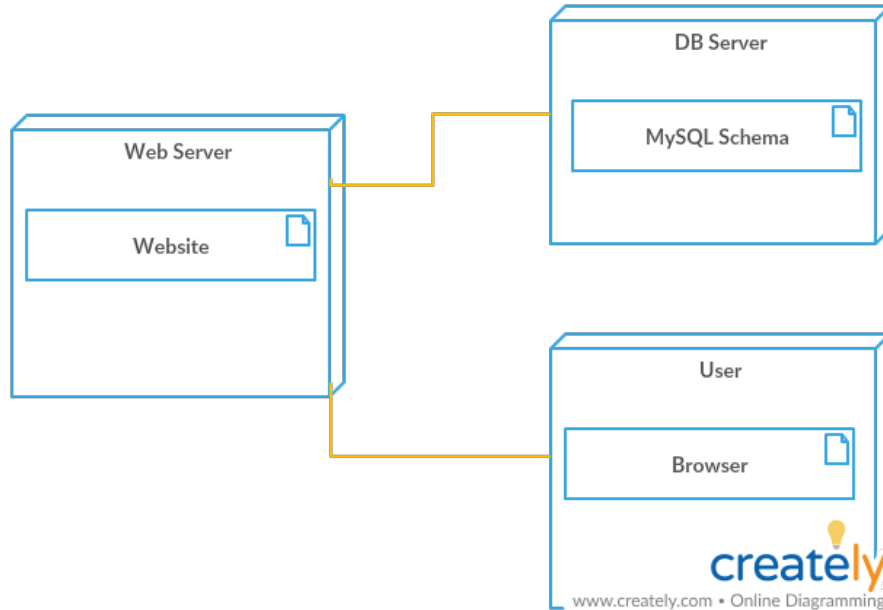
Illustration 13: Deployment Diagram

6. Traceability to Requirements

	UC 1.0	UC 1.1	UC 1.2	UC 1.9	UC 1.11	UC 1.12
US 1	✓					
US 2		✓				
US 4						✓
US 7			✓			
US 8					✓	
US 15				✓		