Detailed Design

for

Tourist Assistor System (TAssist)

Version 1.0 approved

Prepared by Naeema Binthe Ashraf & Mahfujur Rahman Afnan

Khulna University

25 October,2018

# ER Diagram

This is the ER diagram of our project done by MySQL Workbench.

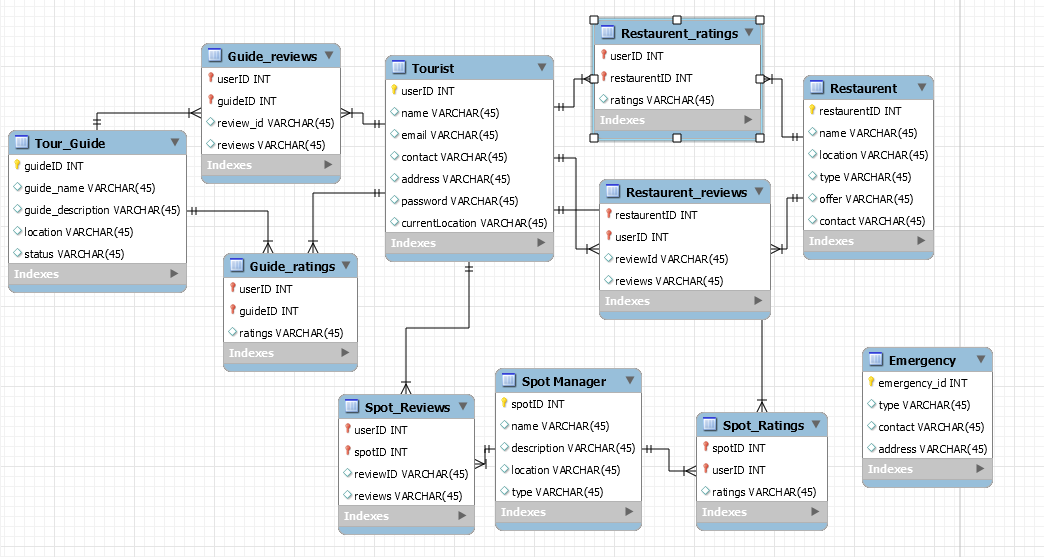
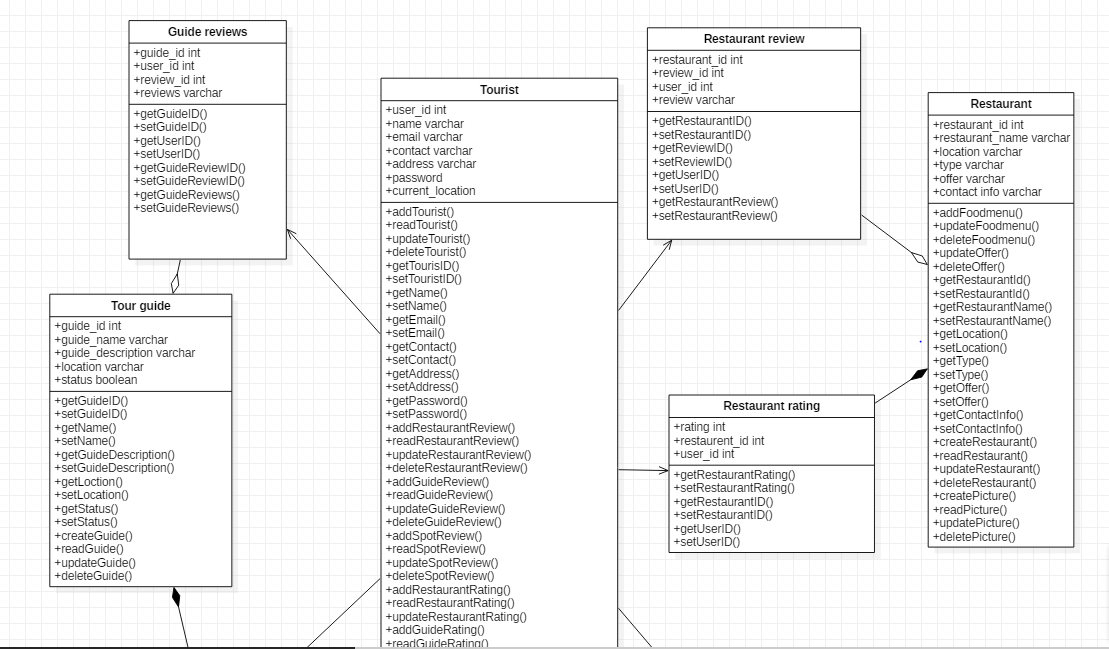
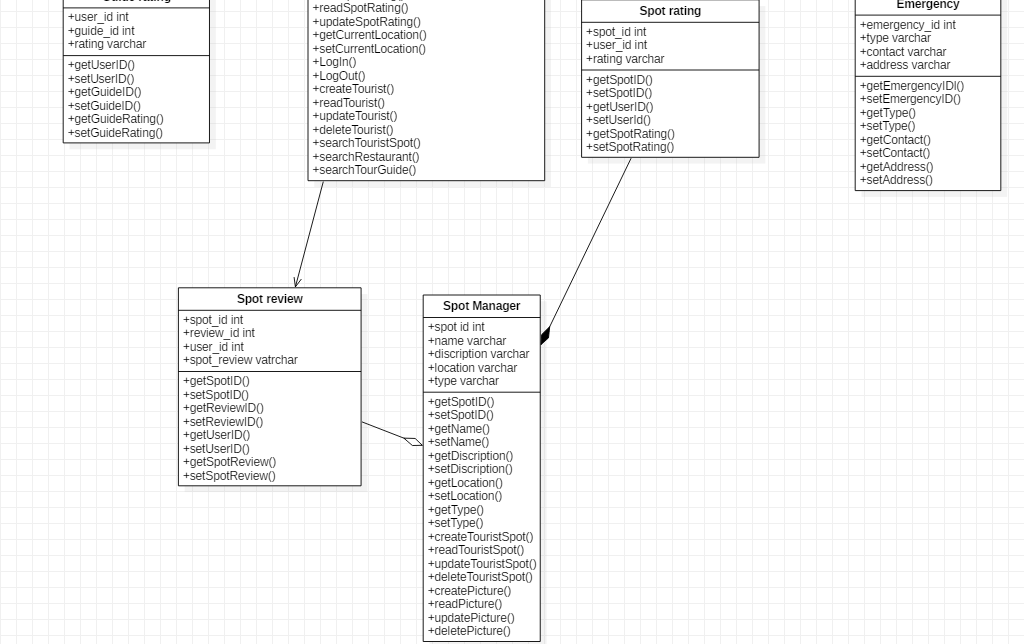


Figure: ER Diagram

# Class Diagram



Figure: UML Diagram

# Class Diagram Details

**Tourist:**

boolean logIn(String ID)

boolean createTourist (Tourist)

Tourist readTourist(String ID)

boolean updateTourist(Tourist)

boolean deleteTourist(Tourist)

Restaurant searchRestaurantbyID(String ID)

List<Restaurant> searchRestaurantbyArea(String location)

TourGuide searchTourGuidebyID(String ID)

List<TourGuide> searchTourGuidebyArea(String location)

TouristSpot searchTouristSpotbyID(String ID)

List<TouristSpot> searchTouristSpotbyArea(String location)

RestaurantReview addRestaurantReview(int user\_ID)

RestaurantReview readRestaurantReview(int user\_ID)

boolean updateRestaurantReview(int user\_ID)

boolran deleteRestaurantReview(int user\_ID)

GuideReview addGuideReview(int user\_ID)

GuideReview readRestaurantReview(int user\_ID)

boolean updateRestaurantReview(int user\_ID)

boolean deleteRestaurantReview(int user\_ID)

TouristSpotReview addSpotReview(int user\_ID)

TouristSpotReview readSpotReview(int user\_ID)

boolean updateSpotReview(int user\_ID)

boolean delete Tourist Spot Review(int user\_ID)

RestaurantRating addRestaurantRating(int user\_ID)

RestaurantRating readRestaurantRating(int user\_ID)

boolean updateRestaurantRating (int user\_ID)

GuideRating addGuideRating(int user\_ID)

GuideRating readRestaurantRatingw(int user\_ID)

boolean updateRestaurantRating(int user\_ID)

TouristSpotRating addSpotRating(int user\_ID)

TouristSpotRating readSpotRating(int user\_ID)

boolean Rating updateSpotRating(int user\_ID)

Restaurant readRestaurant(int restaurant\_id)

TourGuide readTourGuide(int guide\_id)

TouristSpot readTouristSpot(int spot\_id)

boolean logout(String ID)

**Restaurant:**

boolean logIn(String ID)

boolean createRestaurant (Restaurant)

Restaurant readRestaurant(String restaurant\_id)

boolean updateRestaurant(Restaurant)

boolean deleteRestaurant(Restaurant)

RestaurantReview readRestaurantReview(int restaurant\_id)

RestaurantRating readRestaurantRating(int restaurant\_id)

**TouristSpot:**

boolean createTouristSpot (Tourist\_spot)

TouristSpot readTouristSpot (String spot\_id)

boolean updateTouristSpot (Tourist\_Spot)

boolean deleteTouristSpot (Tourist\_Spot)

SpotReview read Spot Review(int Spot \_id)

SpotRating read Spot Rating(int Spot \_id)

**TourGuide:**

boolean logIn(String ID)

boolean createTourGuide (guide)

TourGuide readTourGuide (String guide\_id)

boolean updateTourGuide (Tour\_guide)

boolean deleteTourGuide (Tour\_guide)

GuideReview read Guide Review(int guide \_id)

GuideRating read Guide Rating(int guide \_id)

# Database Script

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema sqltas

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema sqltas

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `sqltas` DEFAULT CHARACTER SET utf8 ;

USE `sqltas` ;

-- -----------------------------------------------------

-- Table `sqltas`.`emergency`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`emergency` (

`emergency\_id` INT(11) NOT NULL,

`type` VARCHAR(45) NULL DEFAULT NULL,

`contact` VARCHAR(45) NULL DEFAULT NULL,

`address` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`emergency\_id`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`tour\_guide`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`tour\_guide` (

`guideID` INT(11) NOT NULL,

`guide\_name` VARCHAR(45) NULL DEFAULT NULL,

`guide\_description` VARCHAR(45) NULL DEFAULT NULL,

`location` VARCHAR(45) NULL DEFAULT NULL,

`status` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`guideID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`tourist`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`tourist` (

`userID` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`email` VARCHAR(45) NULL DEFAULT NULL,

`contact` VARCHAR(45) NULL DEFAULT NULL,

`address` VARCHAR(45) NULL DEFAULT NULL,

`password` VARCHAR(45) NULL DEFAULT NULL,

`currentLocation` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`userID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`guide\_ratings`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`guide\_ratings` (

`userID` INT(11) NOT NULL,

`guideID` INT(11) NOT NULL,

`ratings` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`userID`, `guideID`),

INDEX `fk\_Tourist\_has\_Tour\_Guide\_Tour\_Guide2\_idx` (`guideID` ASC),

INDEX `fk\_Tourist\_has\_Tour\_Guide\_Tourist1\_idx` (`userID` ASC),

CONSTRAINT `fk\_Tourist\_has\_Tour\_Guide\_Tour\_Guide2`

FOREIGN KEY (`guideID`)

REFERENCES `sqltas`.`tour\_guide` (`guideID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Tourist\_has\_Tour\_Guide\_Tourist1`

FOREIGN KEY (`userID`)

REFERENCES `sqltas`.`tourist` (`userID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`guide\_reviews`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`guide\_reviews` (

`userID` INT(11) NOT NULL,

`guideID` INT(11) NOT NULL,

`review\_id` VARCHAR(45) NULL DEFAULT NULL,

`reviews` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`userID`, `guideID`),

INDEX `fk\_Tourist\_has\_Tour\_Guide\_Tour\_Guide1\_idx` (`guideID` ASC),

INDEX `fk\_Tourist\_has\_Tour\_Guide\_Tourist\_idx` (`userID` ASC),

CONSTRAINT `fk\_Tourist\_has\_Tour\_Guide\_Tour\_Guide1`

FOREIGN KEY (`guideID`)

REFERENCES `sqltas`.`tour\_guide` (`guideID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Tourist\_has\_Tour\_Guide\_Tourist`

FOREIGN KEY (`userID`)

REFERENCES `sqltas`.`tourist` (`userID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`restaurent`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`restaurent` (

`restaurentID` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`location` VARCHAR(45) NULL DEFAULT NULL,

`type` VARCHAR(45) NULL DEFAULT NULL,

`offer` VARCHAR(45) NULL DEFAULT NULL,

`contact` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`restaurentID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`restaurent\_ratings`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`restaurent\_ratings` (

`userID` INT(11) NOT NULL,

`restaurentID` INT(11) NOT NULL,

`ratings` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`userID`, `restaurentID`),

INDEX `fk\_Tourist\_has\_Restaurent\_Restaurent1\_idx` (`restaurentID` ASC),

INDEX `fk\_Tourist\_has\_Restaurent\_Tourist1\_idx` (`userID` ASC),

CONSTRAINT `fk\_Tourist\_has\_Restaurent\_Restaurent1`

FOREIGN KEY (`restaurentID`)

REFERENCES `sqltas`.`restaurent` (`restaurentID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Tourist\_has\_Restaurent\_Tourist1`

FOREIGN KEY (`userID`)

REFERENCES `sqltas`.`tourist` (`userID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`restaurent\_reviews`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`restaurent\_reviews` (

`restaurentID` INT(11) NOT NULL,

`userID` INT(11) NOT NULL,

`reviewId` VARCHAR(45) NULL DEFAULT NULL,

`reviews` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`restaurentID`, `userID`),

INDEX `fk\_Restaurent\_has\_Tourist\_Tourist1\_idx` (`userID` ASC),

INDEX `fk\_Restaurent\_has\_Tourist\_Restaurent1\_idx` (`restaurentID` ASC),

CONSTRAINT `fk\_Restaurent\_has\_Tourist\_Restaurent1`

FOREIGN KEY (`restaurentID`)

REFERENCES `sqltas`.`restaurent` (`restaurentID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Restaurent\_has\_Tourist\_Tourist1`

FOREIGN KEY (`userID`)

REFERENCES `sqltas`.`tourist` (`userID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`spot manager`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`spot manager` (

`spotID` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`description` VARCHAR(45) NULL DEFAULT NULL,

`location` VARCHAR(45) NULL DEFAULT NULL,

`type` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`spotID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`spot\_ratings`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`spot\_ratings` (

`spotID` INT(11) NOT NULL,

`userID` INT(11) NOT NULL,

`ratings` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`spotID`, `userID`),

INDEX `fk\_Spot Manager\_has\_Tourist\_Tourist1\_idx` (`userID` ASC),

INDEX `fk\_Spot Manager\_has\_Tourist\_Spot Manager1\_idx` (`spotID` ASC),

CONSTRAINT `fk\_Spot Manager\_has\_Tourist\_Spot Manager1`

FOREIGN KEY (`spotID`)

REFERENCES `sqltas`.`spot manager` (`spotID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Spot Manager\_has\_Tourist\_Tourist1`

FOREIGN KEY (`userID`)

REFERENCES `sqltas`.`tourist` (`userID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `sqltas`.`spot\_reviews`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `sqltas`.`spot\_reviews` (

`userID` INT(11) NOT NULL,

`spotID` INT(11) NOT NULL,

`reviewID` VARCHAR(45) NULL DEFAULT NULL,

`reviews` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`userID`, `spotID`),

INDEX `fk\_Tourist\_has\_Spot Manager\_Spot Manager1\_idx` (`spotID` ASC),

INDEX `fk\_Tourist\_has\_Spot Manager\_Tourist1\_idx` (`userID` ASC),

CONSTRAINT `fk\_Tourist\_has\_Spot Manager\_Spot Manager1`

FOREIGN KEY (`spotID`)

REFERENCES `sqltas`.`spot manager` (`spotID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Tourist\_has\_Spot Manager\_Tourist1`

FOREIGN KEY (`userID`)

REFERENCES `sqltas`.`tourist` (`userID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

# UI Details

1. Here, user can login by inputting username and password and the category of the user.

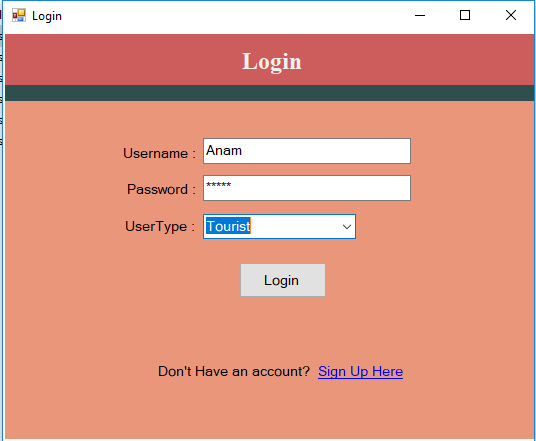


Figure: Login Page

1. If you are not an user, then you have to click Sign Up Here

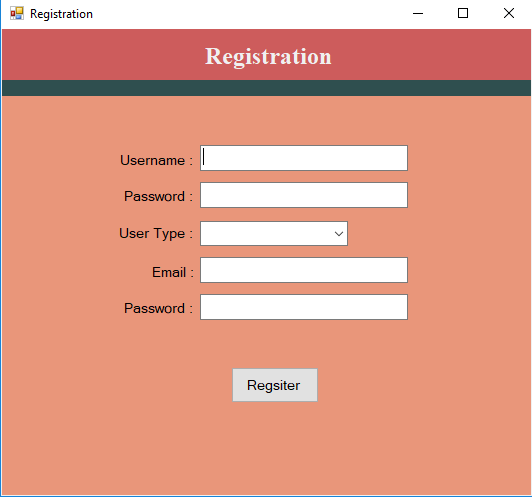


Figure: new user registration form

1. The categories of a user are given in –

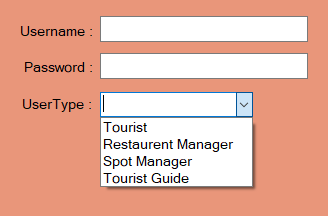


Figure: user categories

1. This is the tourist page

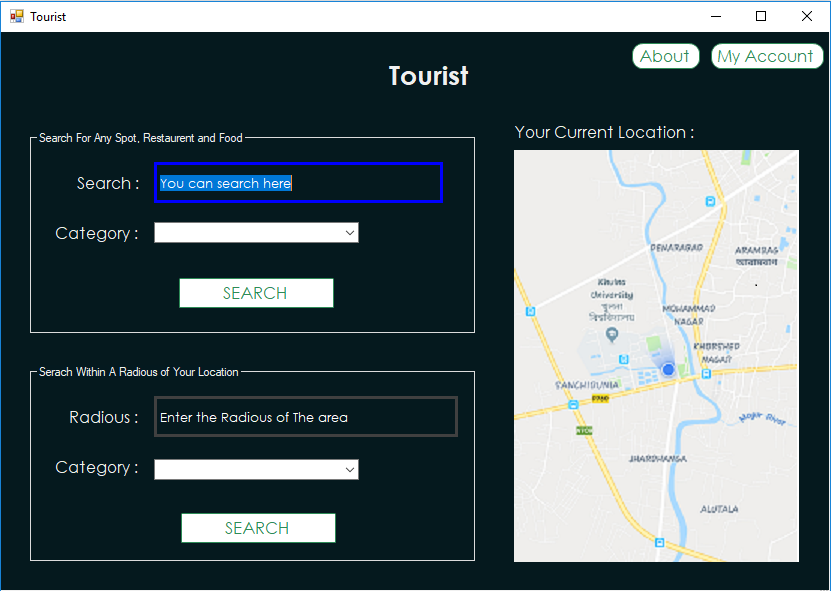


Figure: Tourist Page

1. Tourist can search spots, restaurants, foods from any place by this form

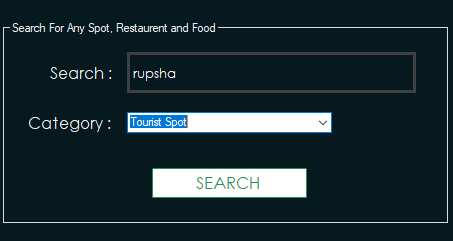


Figure: Tourist Search Option

1. List of all the tourist spots of rupsha

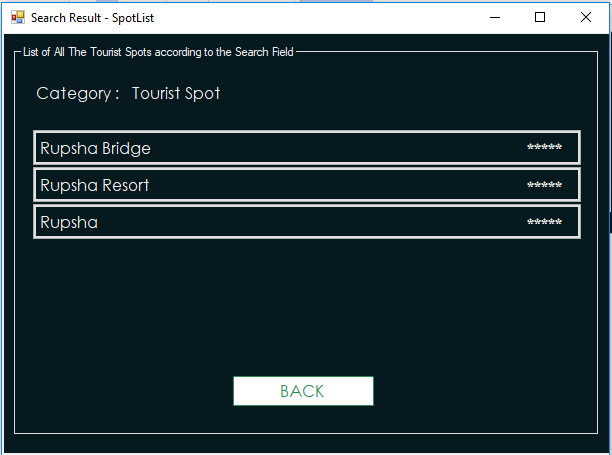


Figure: search results of rupsha

1. The details view of a tourist spot (rupsha bridge)

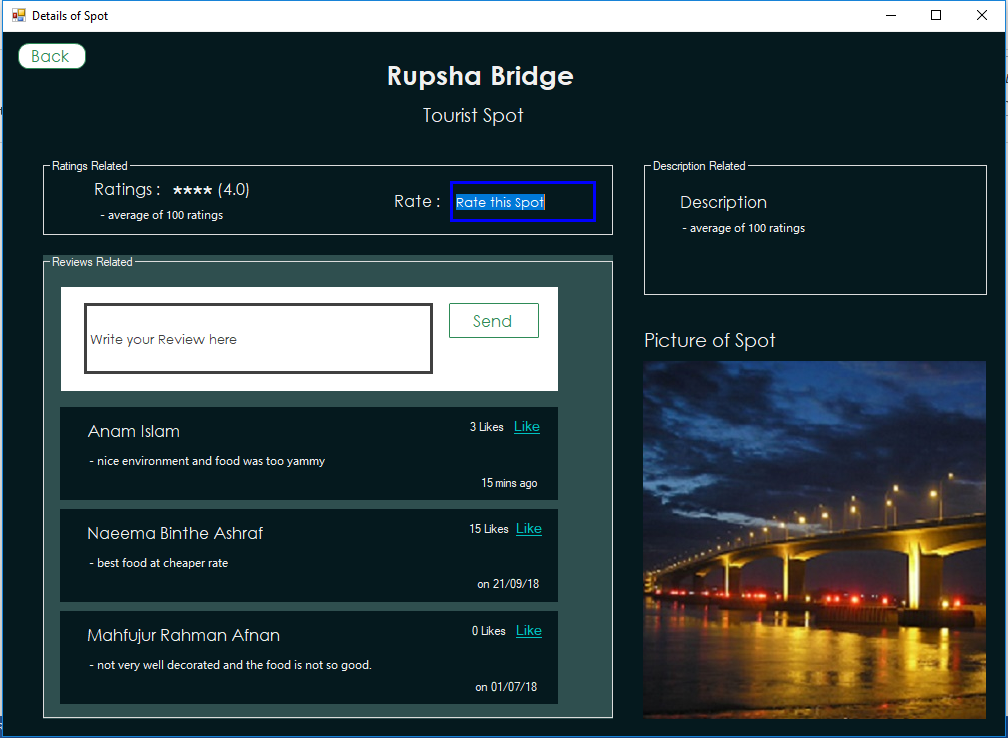


Figure: Rupsha Bridge (Tourist Spot )

1. Search section of tourist in a particular area.

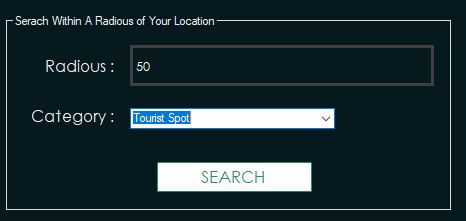


Figure: search option for tourist in his current location among a radius

1. List of all the restaurants according to mezban

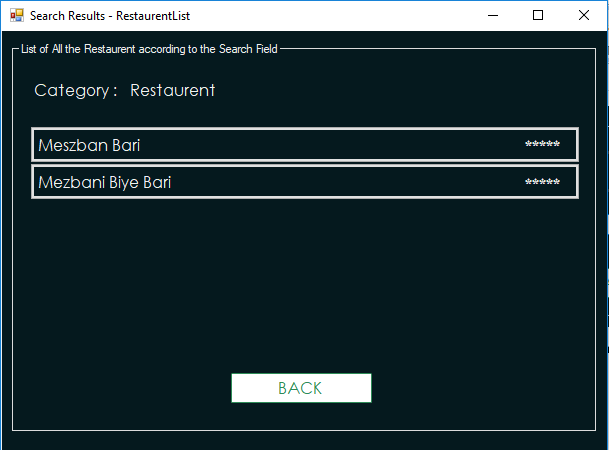


Figure: Search results (Mezban Bari)

1. UI of Mezban bari (View of details of a Restaurant)

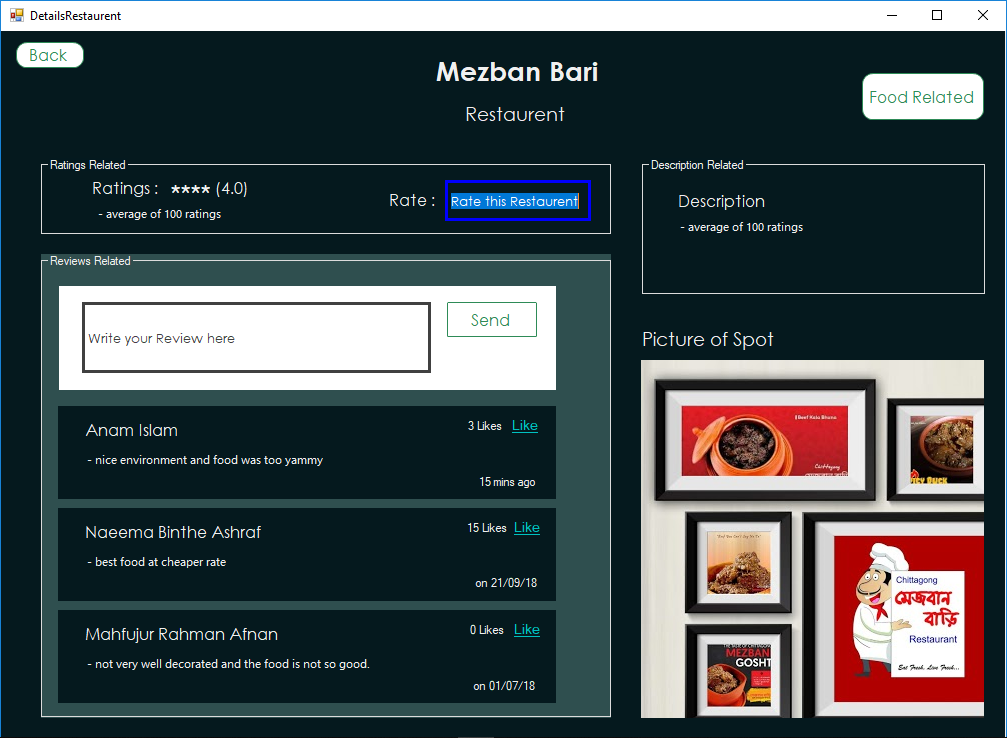


Figure: Details of The restaurant of Mezban Bari

1. The UI of Restaurant Manager is given below.

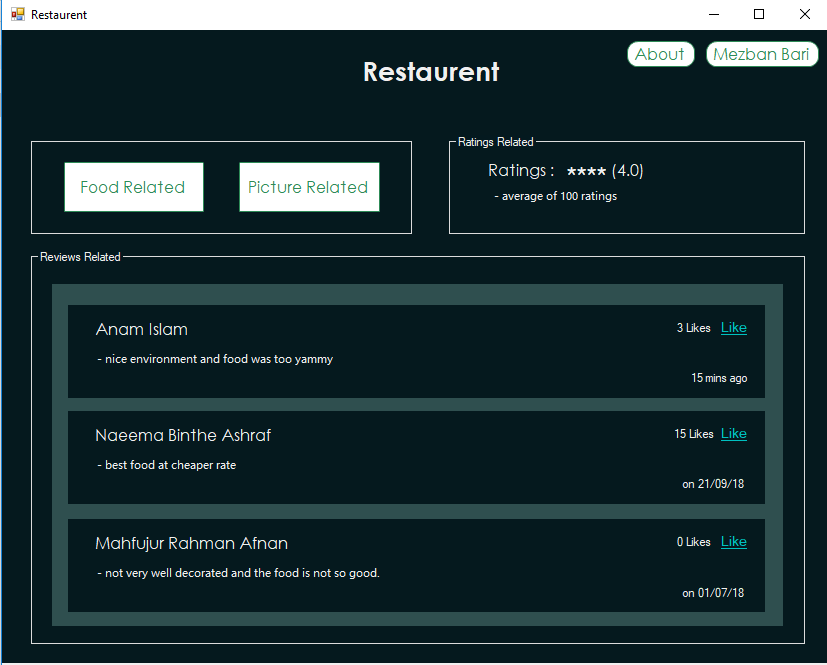


Figure: Restaurant manager page