**Αρχιτεκτονική της Εφαρμογής (3-Επιπέδων)**

**Ροή Εφαρμογής**

1. Ο χρήστης επιλέγει προορισμό, μέρες , τύπο ταξιδιού(εκπαιδευτικό, χαλαρωτικό κλπ),budget(low,medium,high)
2. Ο χρήστης συνδέεται ή εγγράφεται
3. Το frontend στέλνει τα δεδομένα στο TravelPlanServlet.
4. Ο TravelPlanServlet επεξεργάζεται τα δεδομένα, καλεί τις μεθόδους για να δημιουργήσει το πρόγραμμα και το αποθηκεύει στη βάση δεδομένων.
5. Το servlet ανακτά το πρόγραμμα και τα σημεία ενδιαφέροντος.
6. Το frontend εμφανίζει το πρόγραμμα ταξιδιού μαζί με το χάρτη και τις πινέζες.
7. Δυνατότητα δημιουργίας σχολίων

**Χάρτης με Σημεία Ενδιαφέροντος**

Ενσωμάτωση με Google Maps JavaScript API

1. **Presentation Layer (Επίπεδο Παρουσίασης):**
    - **HTML/CSS/JSP**: Για την εμφάνιση της διεπαφής χρήστη, όπου οι χρήστες θα εισάγουν τις πληροφορίες τους (προορισμός, αριθμός ατόμων, ημέρες, προτιμήσεις) και θα βλέπουν το εξατομικευμένο πρόγραμμα ταξιδιού.
2. **Business Logic Layer (Επίπεδο Λογικής Επιχείρησης):**
    - **Servlets**: Υλοποιούν τη λογική για την επεξεργασία των δεδομένων του χρήστη, τη δημιουργία και την ανάκτηση του προγράμματος ταξιδιού από τη βάση δεδομένων.
3. **Data Access Layer (Επίπεδο Πρόσβασης Δεδομένων):**
    - **Βάση Δεδομένων (MySQL)**: Αποθηκεύει τις πληροφορίες σχετικά με τους χρήστες, τα προγράμματα και τα σημεία ενδιαφέροντος.

**2. Servlets**: Δημιουργία servlets που θα χειρίζονται τις αιτήσεις από τον πελάτη (frontend) και θα εκτελούν τη λογική της εφαρμογής.

- **UserServlet**: Διαχειρίζεται τις αιτήσεις για σύνδεση/εγγραφή χρηστών.
- **TravelPlanServlet**: Δημιουργεί το ταξιδιωτικό πρόγραμμα με βάση τις εισόδους του χρήστη.
- Επαλήθευση των στοιχείων σύνδεσης/εγγραφής.
- Δημιουργία και αποθήκευση του προγράμματος ταξιδιού στη βάση δεδομένων.
- Ανάκτηση δεδομένων από τη βάση δεδομένων για εμφάνιση.

**3. DataBase**

Σχέδιο Βάσης Δεδομένων

1. Πίνακας Travellers: Περιέχει στοιχεία των χρηστών.
2. Πίνακας ActivityType: Περιέχει κατηγορίες δραστηριοτήτων.
3. Πίνακας BudgetType: Περιέχει επιλογές budget.
4. Πίνακας TravellerPreferences: Συνδέει τον χρήστη με τους τύπους δραστηριοτήτων και budget.
5. Πίνακας Destinations: Περιέχει τους προορισμούς.
6. Πίνακας Activities: Περιέχει τις δραστηριότητες και συνδέεται με το ActivityType.
7. Πίνακας ActivitySchedule: Περιέχει τα ωράρια κάθε δραστηριότητας.
8. Πίνακας Itinerary: Συνδέει τις δραστηριότητες με ημέρες ταξιδιού για συγκεκριμένο πρόγραμμα.

**-- Πίνακας χρηστών (Travellers)**

```
CREATE TABLE Travellers (
   UserID INT PRIMARY KEY AUTO_INCREMENT,
   UserName VARCHAR(50),
   Password VARCHAR(100)
);

INSERT INTO Travellers (UserName, Email) VALUES
('John Doe', '1234567”),
('Maria Papadopoulou',”123789”);
```

**-- Πίνακας Τύπου Δραστηριότητας (ActivityType)**

```
CREATE TABLE ActivityTypes (
   TypeID INT AUTO_INCREMENT PRIMARY KEY,
   TypeName VARCHAR(50) NOT NULL
);
 -- Εισαγωγή των τεσσάρων τύπων δραστηριοτήτων
INSERT INTO ActivityTypes (TypeName) VALUES
 ('Educational'), ('Adventure'), ('Relaxation'), ('Nature');
```

**-- Πίνακας Budget (BudgetType)**

```
CREATE TABLE BudgetType (
    BudgetID INT PRIMARY KEY AUTO_INCREMENT,
    BudgetName VARCHAR(50)
);

INSERT INTO BudgetType (BudgetName) VALUES
('Low'),
('Medium'),
('High');
```

**-- Πίνακας Προτιμήσεων Χρηστών (TravellerPreferences)**

```
CREATE TABLE TravellerPreferences (
    PreferenceID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT,
    TypeID INT,
    BudgetID INT,
    FOREIGN KEY (UserID) REFERENCES Travellers(UserID),
    FOREIGN KEY (TypeID) REFERENCES ActivityType(TypeID),
    FOREIGN KEY (BudgetID) REFERENCES BudgetType(BudgetID)
);

INSERT INTO TravellerPreferences (UserID, TypeID, BudgetID) VALUES
(1, 1, 2), -- John Doe prefers Museums with Medium budget
(1, 2, 2), -- John Doe prefers Food with Medium budget
(2, 3, 1); -- Maria Papadopoulou prefers Recreation with Low budget
```

**-- Πίνακας Προορισμών (Destinations)**

```
CREATE TABLE Destinations (
    DestinationID INT PRIMARY KEY AUTO_INCREMENT,
    DestinationName VARCHAR(100)
```

```
);

INSERT INTO Destinations (DestinationId,DestinationName) VALUES
(1,'Athens'),
(2,'Thessaloniki');

-- Πίνακας Δραστηριοτήτων (Activities)
CREATE TABLE Activities (
    ActivityID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100),
    DestinationID INT,
    Description TEXT,
    Duration INT, -- Διάρκεια σε λεπτά αν θελετε
    TypeID INT,
    FOREIGN KEY (DestinationID) REFERENCES Destinations(DestinationID),
    FOREIGN KEY (TypeID) REFERENCES ActivityType(TypeID)
);

-- Εισαγωγή δραστηριοτήτων
INSERT INTO Activities (Name, DestinationID ,Description, Duration, TypeID) VALUES
 ('Museum Visit',1, 'Educational visit to the city museum', 120, 1), -- Educational
('Hiking Trail',1, 'Adventure hike in the mountains', 180, 2), -- Adventure
('Beach Relaxation', 'Relaxing time by the beach', 240, 3), -- Relaxation
('Nature Reserve', 'Guided tour in a nature reserve', 150, 4); -- Nature

-- Πίνακας Ωραρίου Δραστηριοτήτων (ActivitySchedule)
CREATE TABLE ActivitySchedule (
    ScheduleID INT PRIMARY KEY AUTO_INCREMENT,
    ActivityID INT,
    StartTime TIME,
    EndTime TIME,
    FOREIGN KEY (ActivityID) REFERENCES Activities(ActivityID)
```

```
);

INSERT INTO ActivitySchedule (ActivityID, StartTime, EndTime) VALUES
(1, '09:00:00', '17:00:00'), -- Acropolis Museum
(2, '12:00:00', '14:00:00'), -- Plaka Food Tour
(3, '18:00:00', '20:00:00'); -- Thermaikos Walk

-- Πίνακας Προγράμματος Ταξιδιού (Itinerary)
CREATE TABLE Itinerary (
    ItineraryID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT,
    ActivityID INT,
    Day INT,
    FOREIGN KEY (UserID) REFERENCES Travellers(UserID),
    FOREIGN KEY (ActivityID) REFERENCES Activities(ActivityID)
);

INSERT INTO Itinerary (UserID, ActivityID, Day) VALUES
(1, 1, 1), -- John Doe's Day 1 Activity: Acropolis Museum
(1, 2, 2), -- John Doe's Day 2 Activity: Plaka Food Tour
(2, 3, 1); -- Maria Papadopoulou's Day 1 Activity: Thermaikos Walk

Πίνακας Παλιών Προγραμμάτων
CREATE TABLE SavedPrograms (
    SavedProgramID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT,
    ActivityID INT,
    Day INT,
    SavedDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (UserID) REFERENCES Travellers(UserID),
    FOREIGN KEY (ActivityID) REFERENCES Activities(ActivityID)
);
```

- SavedItineraryID: Μοναδικός αναγνωριστικός αριθμός.
- UserID: Το ID του χρήστη που έχει αποθηκεύσει το πρόγραμμα.
- ActivityID: Ταυτότητα δραστηριότητας που σχετίζεται με το πρόγραμμα.
- Day: Η ημέρα του προγράμματος.
- SavedDate: Η ημερομηνία που αποθηκεύτηκε το πρόγραμμα.

**ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ**

- Ο Travellers καταγράφει τους χρήστες.
- Ο ActivityType περιλαμβάνει κατηγορίες δραστηριοτήτων.
- Ο BudgetType καθορίζει επίπεδα budget.
- Ο TravellerPreferences συνδέει κάθε χρήστη με πολλαπλές προτιμήσεις δραστηριοτήτων και budget.
- Ο Destinations καθορίζει τους προορισμούς.
- Ο Activities καθορίζει τις δραστηριότητες, συνδέοντάς τες με τον τύπο τους.
- Ο ActivitySchedule ορίζει τα ωράρια για τις δραστηριότητες.
- Ο Itinerary δημιουργεί εξατομικευμένο πρόγραμμα για κάθε χρήστη με συγκεκριμένες δραστηριότητες ανά ημέρα.

**SQL QUERY**

```
SELECT  t.UserName, d.DestinationName, a.Name AS Activity, s.StartTime, s.EndTime, i.Day
FROM  Itinerary i
JOIN  Travellers t ON i.UserID = t.UserID
JOIN  Activities a ON i.ActivityID = a.ActivityID
JOIN   Destinations d ON a.DestinationID = d.DestinationID
JOIN   ActivitySchedule s ON a.ActivityID = s.ActivityID
WHERE    t.UserID = 1 (John Doe)
ORDER BY   i.Day, s.StartTime;
```

**Στόχος να εμφανιστεί κάτι τέτοιο**

**1. Athens, Greece**

- **Day 1:**
    - **09:00 - 11:00:** Visit the Acropolis Museum
    - **11:30 - 13:00:** Lunch at "Kalamaki"
    - **14:00 - 16:00:** Explore the Acropolis
    - **16:30 - 18:30:** Stroll through the Ancient Agora

    - **20:00 - 23:00:** Night out at a Bouzoukia

- **Day 2:**
    - **09:00 - 11:00:** Visit the National Archaeological Museum
    - **11:30 - 13:00:** Lunch at "O Thanasis"
    - **14:00 - 16:00:** Explore Plaka neighborhood
    - **16:30 - 18:30:** Visit Lycabettus Hill for sunset views
    - **20:00 - 22:00:** Dinner at a rooftop restaurant
- **Day 3:**
    - **09:00 - 11:00:** Visit the Temple of Olympian Zeus
    - **11:30 - 13:00:** Lunch at "Strofi"
    - **14:00 - 16:00:** Walk in the National Garden
    - **16:30 - 18:30:** Shopping in Ermou Street
    - **20:00 - 22:00:** Enjoy a cultural show

**Πάμε τώρα στα servlets**

**1. UserServlet για εγγραφή και είσοδο**

```
@WebServlet("/user")
public class UserServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String action = request.getParameter("action");
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        if ("signup".equals(action)) {
            if (registerUser(username, password)) {
                response.getWriter().write("Sign up successful!");
            } else {
                response.getWriter().write("Username already exists.");
            }
        } else if ("login".equals(action)) {
```

```
            if (loginUser(username, password)) {
                response.getWriter().write("Login successful!");
            } else {
                response.getWriter().write("Invalid credentials.");
            }
        }
    }

    private boolean registerUser(String username, String password) {
        // Κώδικας για αποθήκευση νέου χρήστη στη βάση
    }

    private boolean loginUser(String username, String password) {
        // Κώδικας για έλεγχο στοιχείων χρήστη
    }
}
```

**2. Δημιουργία Προγράμματος στο TravelPlanServlet**

Το TravelPlanServlet θα αναζητά δραστηριότητες από τη βάση που να ταιριάζουν με τις προτιμήσεις του χρήστη:

```
@WebServlet("/travelPlan")
public class TravelPlanServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        int userId = Integer.parseInt(request.getParameter("userId"));
        int destinationId = Integer.parseInt(request.getParameter("destinationId"));
        String preferences = request.getParameter("preferences");
        List<Activity> activities = getCustomizedActivities(userId, preferences, destinationId);
        request.setAttribute("activities", activities);
        request.getRequestDispatcher("/viewPlan").forward(request, response);
    }
```

```
    private List<Activity> getCustomizedActivities(int userId, String preferences, int destinationId) {
        // Κώδικας για αναζήτηση δραστηριοτήτων βάσει προτιμήσεων
    }
}
```

**3. Εμφάνιση Προγράμματος στο ViewPlanServlet**

Στο ViewPlanServlet, με χρηση και HTML για την προβολή των δραστηριοτήτων:

```
@WebServlet("/viewPlan")
public class ViewPlanServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        List<Activity> activities = (List<Activity>) request.getAttribute("activities");

        PrintWriter out = response.getWriter();
        out.println("<table>");
        out.println("<tr><th>Ημέρα</th><th>Δραστηριότητα</th><th>Ώρα Έναρξης</th><th>Ώρα Λήξης</th></tr>");
        for (Activity activity : activities) {
            out.println("<tr><td>Day " + activity.getDay() + "</td>");
            out.println("<td>" + activity.getDescription() + "</td>");
            out.println("<td>" + activity.getStartTime() + "</td>");
            out.println("<td>" + activity.getEndTime() + "</td></tr>");
        }
        out.println("</table>");
    }
}
```

**4. Μέθοδος getCustomizedActivities για Προσαρμογή Δραστηριοτήτων**

```
public List<Activity> getCustomizedActivities(int userId, String preferences, int destinationId) {
    List<Activity> activities = new ArrayList<>();
    String[] preferenceList = preferences.split(",");
    String result= "SELECT A.Name, S.StartTime, S.EndTime " +
                "FROM Activities A " …..το sql ερώτημα
")";
```

```
    Try{
        for (int i = 0; i < preferenceList.length; i++) {
            εκτύπωση
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return activities;
}
```