Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Students Kit

Objective

These guidelines are for the student to adopt to make progress in the project.
Given below are the templates for the documents related to the project. These are just guidelines only. The team can improve these.

Requirements Specification (RS)

Following is a template for the RS document. Some example requirements are entered in to it to show how to use the template. Make sure that you enter even the smallest/most trivial requirements also. That would help in validating the system during testing.

No.	Requirement	Essential / Desirable	Description of the Requirement	Remarks
RS1	The system should capture user's facial expressions	Essential	The webcam should capture facial expressions in real-time for emotion analysis	Required for generating emotion-based recommendations
RS2	The system should analyze emotions accurately	Essential	Use a pre-trained deep learning model to analyze facial expressions for emotions	Accuracy of emotion detection is critical for proper recommendations
RS3	The system should recommend music based on detected emotion	Essential	Connect to Spotify API to fetch music suggestions that match detected emotions	Ensures a personalized user experience
RS4	The system should have a user-friendly UI	Essential	Users should easily navigate between pages, view detected emotions, and access recommended tracks	The UI must be consistent across all pages
RS5	The system should provide help screens	Desirable	Store previously detected emotions and recommended songs for each user	**Useful for providing better recommendations over time**

Database Fields Specification

For the user and flights databases, the fields and range of valid values are given below:

No.	Field Name	Range of valid values for the field	Remarks
1	**User ID**	**1 to 1000**	**Unique identifier for each user**
2	**Username**	**Up to 50 characters**	**User's chosen username**
3	**Password**	**Alphanumeric, min 8 characters**	**Password for user authentication**
4	**Approval type**	**Approved/Rejected**	**Miles accepted/rejected by admin**

High Level/Detailed Design (HLD/DD)

Overview of the system

P The Miles Acquisition System is a web-based platform designed to facilitate seamless communication between frequent flyers, airline staff, and administrators. The system captures flight details submitted by users, verifies them through staff, and upon admin approval, credits the corresponding miles to the user's account. The system integrates a responsive user interface (UI), a Spring Boot backend, a MySQL database for persistent storage, and optionally cloud services for scalability and data integrity.

Design Components

Component One: Flight Submission & Miles Request
Purpose: Allows frequent flyer users to submit flight details to earn miles.
Functionality: Form validation, request creation, status set to “Pending” in database.

Component Two: Staff Verification
Purpose: Enables airline staff to verify submitted flight details for authenticity.
Functionality: Reviews user submissions, sets request status to “Verified” or flags for review.

Component Three: Admin Approval & Miles Credit

Purpose: Final approval of miles request and actual credit of miles to user's

account.

Functionality: Updates status to “Approved” or “Rejected” and increments user

miles accordingly in the database.

Pseudocode:

Function submit_flight_details(user_id, flight_info):

```
    // Validate the input flight details
    If not validate_flight_info(flight_info):
        Return "Invalid flight data"
    // Create a new flight request with status 'Pending'
    request = create_request(user_id, flight_info, status="Pending")
    // Save the request to the database
    save_to_database(request)
    Return "Request submitted successfully"
Function verify_flight_request(request_id):
    // Retrieve the request from the database
    request = fetch_request(request_id)
    // Staff reviews the request
    If staff_verifies(request):
        update_status(request_id, "Verified")
    Else
        update_status(request_id, "Flagged")
    Return "Verification complete"
Function approve_request_and_credit_miles(request_id):
    // Retrieve the verified request
    request = fetch_request(request_id)
    If request.status == "Verified":
        // Calculate miles
        miles = calculate_miles(request.flight_info)
```

```
    // Credit miles to user account
    user = fetch_user(request.user_id)
    user.miles += miles
    update_user(user)
    // Mark request as approved
    update_status(request_id, "Approved")
    Return "Miles credited successfully"
Else
    Return "Request not verified"
```

Test-Plan (TP)

The test plan is a list of test cases that need to be run on the system. Ensure the test cases cover all aspects of the system as per the requirements.

No.	Testcase Title	Description	Expected Outcome	The requirement in RS that is being tested	Result
1	**User Login with Valid Credentials**	**Test login using correct email and password**	**User should be logged in and redirected to the appropriate dashboard**	**RS1 (User Authentication)**	**Passed**
2	**Flight Request Submission**	**User submits a flight detail form**	**Recommendations should align genres**	**RS2 (Flight Submission)**	**Passed**
3	**UI Navigation**	**Navigate through all pages (home, recommendations, help)**	**All pages should load without errors**	**RS5**	**Passed**