

EasyTrip Documentation: Refined Sprint

Backlog with Child Issues

This document refines the sprint backlog by breaking down each user story into child issues. These child issues provide detailed tasks for developers to implement the functionality effectively. The backlog is organized into **Epics** and **User Stories**, with each story broken into **Child Issues**.

Epic: EASYTRIP-2

User Story 1: As a user, I want to search for guesthouses by location so that I can find available places easily.

Child Issues:

1. **EASYTRIP-2.1:** Implement a search bar component on the homepage.
 2. **EASYTRIP-2.2:** Develop backend API endpoint (GET `/api/guesthouses?location={location}`) to filter guesthouses by location.
 3. **EASYTRIP-2.3:** Add location-based filtering logic in the backend service.
 4. **EASYTRIP-2.4:** Display search results dynamically on the frontend.
 5. **EASYTRIP-2.5:** Add error handling for invalid or empty search results.
 6. **EASYTRIP-2.6:** Write unit tests for the search functionality (backend and frontend).
-

User Story 2: As an admin, I want to approve or deny the registration request of guesthouse owners.

Child Issues:

1. **EASYTRIP-2.7:** Create an admin dashboard component to display pending guesthouse owner registrations.
2. **EASYTRIP-2.8:** Develop backend API endpoint (GET `/api/admin/pending-owners`) to fetch pending owner registrations.

3. **EASYTRIP-2.9:** Add backend logic to approve (POST `/api/admin/owners/{id}/approve`) or reject (POST `/api/admin/owners/{id}/reject`) owner registrations.
 4. **EASYTRIP-2.10:** Implement frontend buttons for approve/reject actions.
 5. **EASYTRIP-2.11:** Notify guesthouse owners via email upon approval/rejection.
 6. **EASYTRIP-2.12:** Write unit tests for admin approval functionality.
-

User Story 3: As an admin, I want to view all user and guesthouse owner accounts so that I can manage platform users.

Child Issues:

1. **EASYTRIP-2.13:** Create a user management table in the admin dashboard.
 2. **EASYTRIP-2.14:** Develop backend API endpoint (GET `/api/admin/users`) to fetch all users and owners.
 3. **EASYTRIP-2.15:** Add filtering and pagination to the user management table.
 4. **EASYTRIP-2.16:** Implement frontend functionality to display user details (role, email, status).
 5. **EASYTRIP-2.17:** Add backend logic to deactivate or delete users.
 6. **EASYTRIP-2.18:** Write unit tests for user management functionality.
-

User Story 4: As a guesthouse owner, I want to add and update room availability and prices so that customers can see accurate booking options.

Child Issues:

1. **EASYTRIP-2.19:** Create a guesthouse owner dashboard component for managing room availability and pricing.
 2. **EASYTRIP-2.20:** Develop backend API endpoint (PUT `/api/guesthouses/{id}/rooms`) to update room availability and pricing.
 3. **EASYTRIP-2.21:** Add frontend form for updating room details (availability, price, etc.).
 4. **EASYTRIP-2.22:** Implement real-time validation for room updates (e.g., no negative prices).
 5. **EASYTRIP-2.23:** Notify travelers if a booked room becomes unavailable.
 6. **EASYTRIP-2.24:** Write unit tests for room management functionality.
-

User Story 5: As a guesthouse owner, I want to receive notifications when a booking is made so that I can prepare for guests.

Child Issues:

1. **EASYTRIP-2.25:** Implement a notification system for guesthouse owners.
 2. **EASYTRIP-2.26:** Develop backend logic to trigger notifications on new bookings (POST /api/bookings).
 3. **EASYTRIP-2.27:** Add a notifications panel to the guesthouse owner dashboard.
 4. **EASYTRIP-2.28:** Send email notifications to owners for new bookings.
 5. **EASYTRIP-2.29:** Mark notifications as read/unread in the frontend.
 6. **EASYTRIP-2.30:** Write unit tests for the notification system.
-

User Story 6: As a user, I want to cancel or modify my booking so that I have flexibility in my plans.

Child Issues:

1. **EASYTRIP-2.31:** Add a "My Bookings" section to the traveler dashboard.
 2. **EASYTRIP-2.32:** Develop backend API endpoint (PUT /api/bookings/{id}) to modify bookings.
 3. **EASYTRIP-2.33:** Add backend logic to handle booking cancellations (DELETE /api/bookings/{id}).
 4. **EASYTRIP-2.34:** Implement frontend forms for modifying bookings (e.g., change dates).
 5. **EASYTRIP-2.35:** Notify guesthouse owners of booking modifications or cancellations.
 6. **EASYTRIP-2.36:** Write unit tests for booking modification and cancellation.
-

User Story 7: As a user, I want to sign up and log in securely so that I can access the system.

Child Issues:

1. **EASYTRIP-2.37:** Create a secure authentication system using JWT tokens.
2. **EASYTRIP-2.38:** Develop backend API endpoints for user registration (POST /api/auth/register) and login (POST /api/auth/login).
3. **EASYTRIP-2.39:** Implement password hashing and validation.

4. **EASYTRIP-2.40:** Add frontend forms for registration and login.
 5. **EASYTRIP-2.41:** Implement role-based access control (Traveler, Owner, Admin).
 6. **EASYTRIP-2.42:** Write unit tests for authentication functionality.
-

User Story 8: As a guesthouse owner, I want to register my business so that travelers can find and book my guesthouse.

Child Issues:

1. **EASYTRIP-2.43:** Create a guesthouse registration form in the frontend.
 2. **EASYTRIP-2.44:** Develop backend API endpoint (POST /api/guesthouses) to handle guesthouse registration.
 3. **EASYTRIP-2.45:** Add validation for guesthouse details (e.g., name, description, amenities).
 4. **EASYTRIP-2.46:** Implement image upload functionality for guesthouse photos.
 5. **EASYTRIP-2.47:** Notify admins of new guesthouse registration requests.
 6. **EASYTRIP-2.48:** Write unit tests for guesthouse registration.
-

User Story 9: As a user, I want to view guesthouse details and amenities so that I can make an informed decision.

Child Issues:

1. **EASYTRIP-2.49:** Create a guesthouse details page in the frontend.
 2. **EASYTRIP-2.50:** Develop backend API endpoint (GET /api/guesthouses/{id}) to fetch guesthouse details.
 3. **EASYTRIP-2.51:** Display guesthouse amenities, pricing, and availability on the details page.
 4. **EASYTRIP-2.52:** Add a booking form to the guesthouse details page.
 5. **EASYTRIP-2.53:** Implement real-time availability checks.
 6. **EASYTRIP-2.54:** Write unit tests for guesthouse details functionality.
-

Workflow Overview

1. **Traveler Workflow**
 - Search for guesthouses → View details → Book → Modify/Cancel booking.

2. Guesthouse Owner Workflow

- Register guesthouse → Update room details → Manage bookings → Receive notifications.

3. Admin Workflow

- Approve/reject guesthouse owners → View all users → Manage guesthouse listings.
-

Error Handling

- **Invalid Input:** Display error messages for invalid search queries, registration details, or booking modifications.
- **Unauthorized Access:** Redirect users to the login page if they attempt to access restricted areas.
- **Server Errors:** Display a generic error message and log the issue for debugging.

EasyTrip Website Workflow and Features

Report

Overview

EasyTrip is a web-based platform designed to connect travelers with guesthouse owners. The platform allows travelers to search for guesthouses, view details, and book stays, while guesthouse owners can list their properties, manage bookings, and interact with travelers. The application is built using **Spring Boot** for the backend and **ReactJS** for the frontend. Currently, the platform supports **cash payments** only.

Workflow

1. Traveler Workflow

Step 1: Search for Guesthouses

- Travelers visit the homepage and use the search bar to find guesthouses by location.
- The system displays a list of guesthouses matching the search criteria.

Step 2: View Guesthouse Details

- Travelers click on a guesthouse to view detailed information, including amenities, pricing, and availability.
- The guesthouse details page also displays images and a description.

Step 3: Book a Guesthouse

- Travelers select check-in and check-out dates and confirm the booking.
- The system validates the availability and creates a booking.

Step 4: Modify or Cancel Booking

- Travelers can modify or cancel their bookings through the "My Bookings" section.
- The system updates the booking status and notifies the guesthouse owner.

Step 5: Payment

- Travelers pay in cash upon arrival at the guesthouse.
-

2. Guesthouse Owner Workflow

Step 1: Register Guesthouse

- Owners register their guesthouse by filling out a form with details such as name, description, amenities, and pricing.
- The system sends the registration request to the admin for approval.

Step 2: Manage Listings

- Approved owners can update room availability, pricing, and other details through their dashboard.
- Owners can also upload images and modify descriptions.

Step 3: Manage Bookings

- Owners receive notifications for new booking requests.
- Owners can accept or reject bookings through their dashboard.

Step 4: Notifications

- Owners receive notifications for new bookings, modifications, and cancellations.
-

3. Admin Workflow

Step 1: Approve/Reject Guesthouse Owners

- Admins review guesthouse owner registration requests and approve or reject them.
- Approved owners can list their guesthouses, while rejected owners receive feedback.

Step 2: Manage Users

- Admins can view and manage all user accounts (travelers and owners).
- Admins can deactivate or delete accounts if necessary.

Step 3: Manage Guesthouse Listings

- Admins can view all guesthouse listings and approve/reject new listings.
 - Admins can remove listings that violate platform policies.
-

Features

1. Traveler Features

Search and Filter

- Search for guesthouses by location.
- Filter results by price range, amenities, and availability.

Guesthouse Details

- View detailed information about guesthouses, including amenities, pricing, and images.
- Check real-time availability.

Booking Management

- Book guesthouses by selecting dates and confirming the booking.
- Modify or cancel bookings through the "My Bookings" section.

User Authentication

- Secure sign-up and login functionality.
 - Role-based access control (Traveler, Owner, Admin).
-

2. Guesthouse Owner Features

Guesthouse Registration

- Register a new guesthouse by providing details such as name, description, amenities, and pricing.
- Upload images and set room availability.

Listing Management

- Update guesthouse details, including pricing and availability.
- Manage bookings (accept/reject).

Notifications

- Receive notifications for new bookings, modifications, and cancellations.
-

3. Admin Features

User Management

- View and manage all user accounts (travelers and owners).
- Deactivate or delete accounts.

Guesthouse Approval

- Approve or reject guesthouse owner registration requests.
- Approve or reject new guesthouse listings.

Reporting and Moderation

- Handle complaints and reports from users.
 - Remove listings that violate platform policies.
-

Payment Workflow

Cash Payment

- 1. Booking Confirmation**
 - Travelers book a guesthouse and receive a confirmation with booking details.
- 2. Payment Upon Arrival**
 - Travelers pay in cash when they check in at the guesthouse.
- 3. Confirmation to Owner**
 - The system notifies the guesthouse owner of the booking and payment details.

Error Handling

Common Errors

- **Invalid Search Query:** Display error message for invalid or empty search results.
- **Unauthorized Access:** Redirect users to the login page if they attempt to access restricted areas.
- **Booking Errors:** Display error messages for unavailable dates or invalid booking modifications.
- **Server Errors:** Display a generic error message and log the issue for debugging.