# Obtayn

Souvenir Exchange

A lot of times, there are region-specific authentic goods that we cannot always buy from e-commerce websites. The best way to procure these goods is to ask an acquaintance to bring them for you when they return. However, it is not necessary that everytime we need something, we have someone to bring it for us.

To facilitate this exchange, we thought of a common platform where users post requests and the ones travelling can accept them. These requests can be viewed in the feed, along with photos of required items. Travellers can directly accept requests from here, as well as post their travel plans. Once a request gets accepted, the requestor & traveller can chat with each other, and set up payment via the application itself.

## Requirements

1. Travellers can post their travel plans.
2. Travellers can post available items.
3. Requestor can pay the Traveller via the application.

## 

## User Stories

Write intro

* I, as a Traveller, want to post my travel plans
* I, as a Requestor, want to send a Request from the Feed (Travel plan post).
* I, as a Traveller, want to accept targeted request from incoming request
* I, as a Traveller, want to post an available item. (Available Post)
* I, as a Requestor, want to send a Request via Available Post.

### Post Travel Plans

|  |  |
| --- | --- |
| **1** |  |
| **User Story** | I, as a Traveller, want to post my travel plans |
| **Description** | When I click on the “Create Post” button in the header, or the one above the Feed, I see a pop-up window. When I click on “Tag”, I see a drop-down list containing Request, Post, Travel Plans, and I select “Travel Plans”.  The pop-up window contains a form with various fields like Title, Message, Location & Dates (Start date & End date).  When I click on the “Add Plans” button, the pop-up window closes and my Post appears at the top of the Feed. System shows my Username and Profile picture on the left of the post title. |

|  |  |
| --- | --- |
| Task | Estimated Units |
| Use “Create a Post” component from story #7. | 0 |
| Make a component for request form containing :   * Title * Message * Location * Start date * End date * Add Plan button | 4 |
| Write a function to trigger the modal on click for Create Post component. | 1 |
| Write a function to validate whether form fields are filled. | 1 |
| Write a function to show request form, appreciation form or travel plan form depending on Tag | 0 |
| Write a function to validate Start date & End date. | 3 |
| Write a function to add the new Travel Plan when the Add Plan button is clicked. | 2 |
| Write a function to show the new post on top of the Feed. | 2 |
| Write a function to handle “Network Error” exception. | 2 |
| Add Travel Plan component to Feed containing   * User’s name * Profile picture * Title * Message * Timestamp * Location * Start & End date * Send Request button | 4 |
| **Total** | **19** |

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Actual Result |
| User clicks “Create Post” button | System shows the Create Post modal. |  |
| User clicks the “Add Plan” button. | The post is added to the database and to the Feed. |  |
| Required fields are empty.  (title, message, location , start & end date) | System prompts Users to fill fields that are marked required. |  |
| User clicks outside the modal or clicks close button | The modal is closed |  |
| Network error occurs before Users’ post is uploaded. | System should alert Users of the issue, and upload the post as soon as connection is established. |  |
| User tries to pick a date previous to current day as the start date. | System disables previous dates in the date picker. |  |
| User tries to pick a date previous to the start date as the end date. | System disables dates previous to selected start date in the date picker. |  |
| User enters invalid location | System prompts User to enter a valid location. |  |

## 

### Send request from Travel Plan

|  |  |
| --- | --- |
| **2** |  |
| **User Story** | I, as a Requestor, want to send a Request from the Feed. |
| **Description** | When I see a Travel Plan having a location which I want an item from, I click on the Send Request button in that post. This opens the Create Post pop-up modal with the Category, Location & Title autofilled and “Request” as the Tag.  This window contains a form with various fields like Message & Photo. Photo is an optional field. I cannot upload more than 3 photos.  On clicking the “Add Request” button, the pop-up window closes and my Request is sent to the Traveller. The request is added to my Outgoing Requests. |

|  |  |
| --- | --- |
| Task | Estimated Units |
| Write a function to   * Open Create Post modal when the “Send Request” button is clicked. * Set Tag as Request * Set location as that of the Travel Plan location | 5 |
| Use “Create a Post” component from story #7. | 0 |
| Make a component for request form containing :   * Title * Message * Location (autofilled) * Category * images(if any, max 3) * Add Request button | 2 |
| Write a function to validate whether form fields are filled. | 1 |
| Write a function to validate image constraints. | 0 |
| Write a function to   * Add the new Targeted Request when the “Send Request” button is clicked. * Close Create Post modal * Set “request\_receiver” attribute as Traveller * Set Targeted Request status to Pending * Add Request to Travellers Incoming Requests. * Add the request to requestor’s outgoing request. | 8 |
| Write a function to handle “Network Error” exception. | 1 |
| **Total** | **17** |

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Actual Result |
| User clicks “Send Request” button | System shows the Create Post modal. |  |
| User clicks the “Add Request” button. | The Targeted Request is added to the database. |  |
| Required fields are empty.  (title, message, category) | System prompts Users to fill fields that are marked required. |  |
| User clicks outside the modal or clicks close button | The modal is closed |  |
| Travellers try to send requests to themselves. | System disables the “Send Request” button when Users view their own posts. |  |
| Network error occurs before Users’ post is uploaded. | System should alert Users of the issue, and upload the post as soon as connection is established. |  |

## 

### Accept a targeted Request

|  |  |
| --- | --- |
| **3** |  |
| **User Story** | I, as a Traveller, want to accept targeted request from incoming request |
| **Description** | When I open my requests tab, under incoming requests I can accept or decline the request .  When I click accept, the request is highlighted. The User is added to my Chat list, and we can start a new conversation.  When I decline the request is removed from my incoming requests. |

|  |  |
| --- | --- |
| Task | Estimated Units |
| Create targeted requests component with details   * Requestor * Title * Timestamp * Location * message * images(if any) * accept request button * decline request button | 8 |
| Make a function to handle accept request   * adds a notification. * change the request status from targeted to accepted request * add Requestor to the Traveller’s Chat list. * highlight current Request. | 4 |
| Write a function to handle decline request   * Permanently delete the request * Refresh the request feed * Add notification of deleted request saying “The traveller is unable to fulfil this request as of right now”. | 4 |
| **Total** | **16** |

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Actual Result |
| User accepts a Targeted Request from the incoming feed. | The system sends a notification to the requestor.  The request style is changed to accepted and is highlighted. |  |
| Network error occurs while User is accepting a Request. | System notifies the User about the issue, and changes Request status as soon as connection is established. |  |
| User declines a Targeted request. | The system deletes the request,  And refreshes the feed |  |
| The system sends a notification to the requestor. |  |

### Post available items

|  |  |
| --- | --- |
| **4** |  |
| **User Story** | I, as a Traveller, want to post available items. |
| **Description** | When I click on the “Create Post” button in the header, or the one above the Feed, I see a pop-up window. When I click on “Tag”, I see a drop-down list containing Request, Post, Travel Plans, Available Items and I select “Available Items”.  The pop-up window contains a form with various fields like Title, Message, Category, Location, Images.  When I click on the “Add Items” button, the pop-up window closes and my Post appears at the top of the Feed. System shows my Username and Profile picture on the left of the post title. |

|  |  |
| --- | --- |
| Task | Estimated Units |
| Use “Create a Post” component from story #7. | 0 |
| Make a component for Available Items form containing :   * Title * Message * Location * Images * Add Items button | 4 |
| Use function to trigger the modal on click for Create Post component. | 0 |
| Use function to validate whether form fields are filled. | 0 |
| Use function that shows request form, appreciation form, travel plan or available items form depending on Tag | 0 |
| Use function to validate image constraints (size, extension, number of images) | 1 |
| Write a function to add the new Available Items when the Add Items button is clicked. | 2 |
| Use function to show the new post on top of the Feed. | 0 |
| Use function to handle “Network Error” exception. | 0 |
| Add Available Item component to Feed containing   * User’s name * Profile picture * Title * Message * Timestamp * Location * Images * Send Request button | 4 |
| **Total** | **11** |

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Actual Result |
| User clicks “Create Post” button | System shows the Create Post modal. |  |
| User clicks the “Add Items” button. | The post is added to the database and to the Feed. |  |
| Required fields are empty.  (title, message, location, images) | System prompts Users to fill fields that are marked required. |  |
| User clicks outside the modal or clicks close button | The modal is closed |  |
| Network error occurs before Users’ post is uploaded. | System should alert Users of the issue, and upload the post as soon as connection is established. |  |
| User enters invalid location | System prompts User to enter a valid location. |  |

### 

### Request for available item

|  |  |
| --- | --- |
| **5** |  |
| **User Story** | I, as a Requestor, want to send a Request from Available Items post. |
| **Description** | When I see an Available Items post having a location which I want an item from, I click on the Send Request button in that post.  On clicking the “Send Request” button, the pop-up window closes and my Request is sent to the Traveller. The request is added to my Outgoing Requests. |

|  |  |
| --- | --- |
| Task | Estimated Units |
| Write a function to   * Open Create Post modal when the “Send Request” button is clicked. * Set Tag as Request * Set location as that of the Travel Plan location | 5 |
| Use “Create a Post” component from story #7. | 0 |
| Make a component for request form containing :   * Title (autofilled) * Message * Location (autofilled) * Send Request button | 2 |
| Use function to validate whether form fields are filled. | 0 |
| Write a function to   * Add the new Targeted Request when the “Add Request” button is clicked. * Close Create Post modal * Set “request\_receiver” attribute as Traveller * Set Targeted Request status to Pending * Add Request to Travellers Incoming Requests. * Add the Request to Requestor’s Outgoing Requests. | 8 |
| Use function to handle “Network Error” exception. | 0 |
| **Total** | **15** |

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Actual Result |
| User clicks “Send Request” button | System shows the Create Post modal. |  |
| User clicks the “Add Request” button. | The Targeted Request is added to the database. |  |
| Message field is marked required, but is empty. | System prompts Users that the Message field cannot be empty. |  |
| User clicks outside the modal or clicks the close button. | The modal is closed. |  |
| Users try to send requests to themselves. | System disables the “Send Request” button when Users view their own posts. |  |
| Network error occurs before Users’ Request is sent. | System should alert Users of the issue, and send the Request as soon as connection is established. |  |

## 

### **Total time estimated** = 78 units

= 34 hours

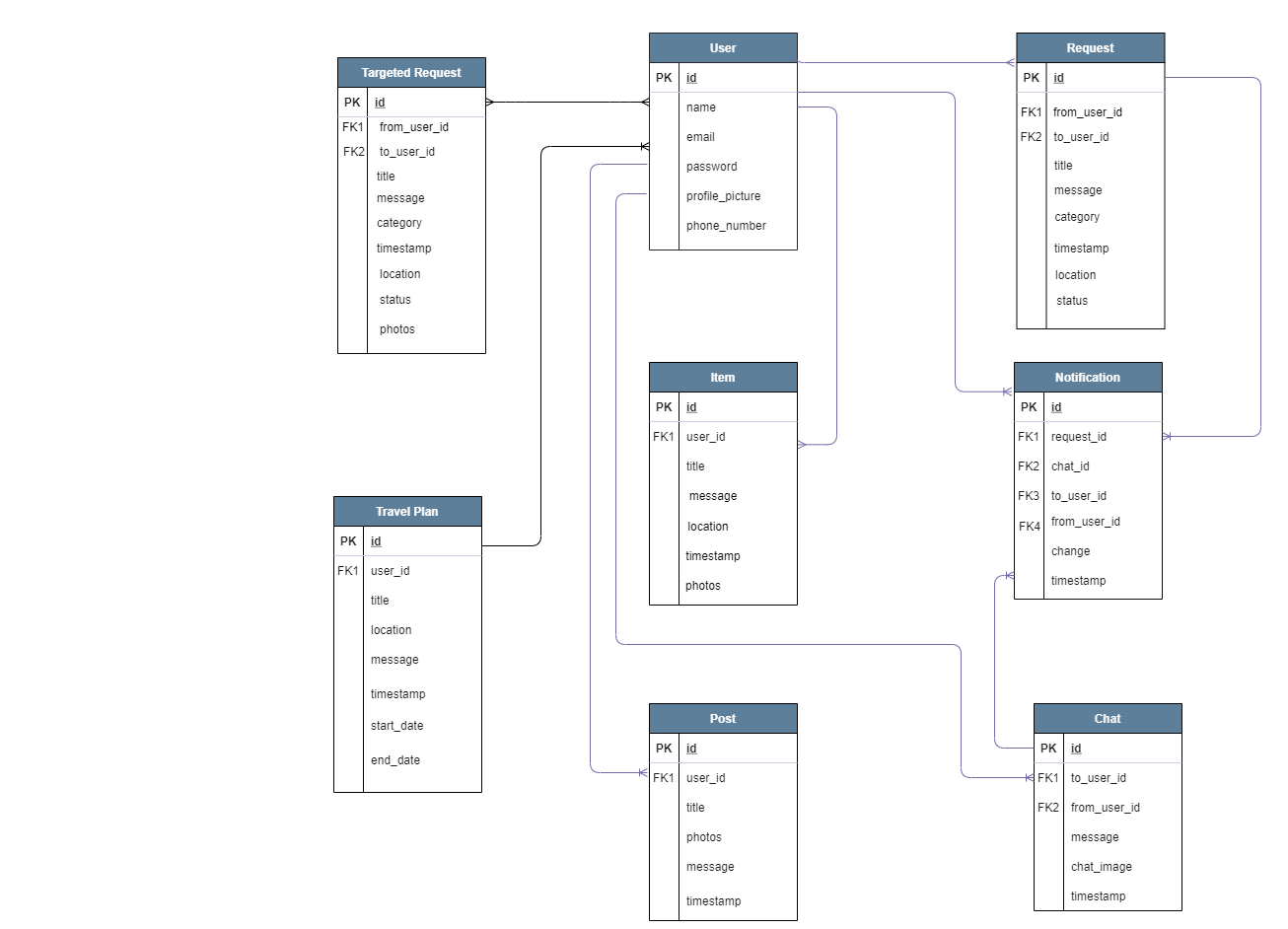
≅ 4 days

## 

## Database Design

### Entity Relationship diagram

* This diagram shows relationships between the entities in our system.
* A User can post about multiple Items, hence User-Item is a one-many relationship.
* A User can post multiple travel plans , hence User-Travel\_plan is one-many relationship.
* A User can send multiple targeted\_request ,hence User-Targeted\_request is many to many.



## 

### 

### Entity Description

**USER**

This database table stores User details. Each row is identifiable via the id, which is used as primary key and is unique for each User. Profile\_picture & phone\_number are optional fields for Users.

|  |  |
| --- | --- |
| Attribute | Constraints |
| id | PRIMARY KEY |
| name | NOT NULL |
| password | NOT NULL |
| profile\_picture |  |
| phone\_number |  |

## 

**REQUEST**

This table represents a Request posted by User(s). “to\_user\_id” is NULL by default, and it changes only when that Request is accepted by another User.

Each Request has a title (identifies a Request), message (requirement description written by the User), category (Clothing, Home & Living, Hobbies, Electronics, Food), timestamp (date & time when the Request is posted), location (the place where User)

|  |  |
| --- | --- |
| Attribute | Constraints |
| id | PRIMARY KEY |
| to\_user\_id | NOT NULL |
| from\_user\_id | NOT NULL |
| title |  |
| message |  |
| category |  |
| timestamp |  |
| location |  |
| photos |  |
| status |  |

**TARGETED REQUEST**

This table represents a Targeted Request sent by User(s). “to\_user\_id” is that of the Traveller who posted their travel plan .The status is pending by default it changes only when that Request is accepted by the Traveller.

Each Request has a title (identifies a Request), message (requirement description written by the User), category (Clothing, Home & Living, Hobbies, Electronics, Food), timestamp (date & time when the Request is posted), location which is autofilled from the Travel plans post.

|  |  |
| --- | --- |
| Attribute | Constraints |
| id | PRIMARY KEY |
| to\_user\_id | NOT NULL |
| from\_user\_id | NOT NULL |
| title |  |
| message |  |
| category |  |
| timestamp |  |
| location |  |
| photos |  |
| status |  |

**POST**

Represents who(user\_id) has posted what(title, photos, message) on the Feed. Also stores the time when it was posted (timestamp). Users may or may not add photos to their posts, so it is an optional field.

|  |  |
| --- | --- |
| Attribute | Constraints |
| id | PRIMARY KEY |
| user\_id | NOT NULL |
| title |  |
| message |  |
| timestamp |  |
| photos |  |

**TRAVEL PLAN**

Represents travellers(user\_id) who have posted a travel plan(title, message, start\_date, end\_date, location) on the Feed. Also stores the time when it was posted (timestamp).

|  |  |
| --- | --- |
| Attribute | Constraints |
| id | PRIMARY KEY |
| user\_id | NOT NULL |
| title |  |
| message |  |
| timestamp |  |
| location |  |
| start \_date |  |
| end\_date |  |

**CHAT**

Represents each message that Users send to each other. It contains 2 foreign keys - sender (from\_user\_id) & receiver (to\_user\_id). Users can send just a message (chat\_image is NULL) or Users can directly send an image without any message (message), which makes them optional fields.

|  |  |
| --- | --- |
| Attribute | Constraints |
| id | PRIMARY KEY |
| from\_user\_id | NOT NULL |
| to\_user \_id |  |
| message |  |
| timestamp |  |

**NOTIFICATION**

A Notification can be triggered by 2 events - User receives a message from another User (to\_user\_id, message) or User’s request is accepted by another User (request\_id, change). Since notifications are temporary, it also stores the time when notification appears (timestamp).

|  |  |
| --- | --- |
| Attribute | Constraints |
| id | PRIMARY KEY |
| trigger\_event | NOT NULL |
| trigger\_event\_id | NOT NULL |
| from\_user\_id | NOT NULL |
| to\_user \_id | NOT NULL |
| message |  |
| timestamp |  |

**ITEM**

Represents an available item that the Traveller has posted on the Feed.

Also stores the time when it was posted (timestamp). Photos are a required field.

|  |  |
| --- | --- |
| Attribute | Constraints |
| id | PRIMARY KEY |
| user\_id | NOT NULL |
| title |  |
| message |  |
| location |  |
| timestamp |  |
| photos |  |

## 

## Component Diagram

## 

## Screen Mockups

