CampusRecover - Design Document

CS5610 Web Development - Spring 2026 Authors: Preshit Ravindra Pimple & Gurudatt Pramod Gaonkar

---

1. Project Description

Overview

CampusRecover is a campus-wide lost and found coordination platform that helps Northeastern University students recover lost items through structured reporting and intelligent matching. Every semester, students lose valuable items like laptops, keys, wallets, and AirPods around campus. Current solutions rely on scattered group chats, buried emails, and physical lost-and-found boxes spread across campus buildings that students must visit individually.

Key Features

- **Lost Item Reporting** - Students can submit detailed lost item reports with description, last seen location, date/time, and contact info
- **Found Item Reporting** - Good Samaritans can report found items with pickup location and holding details
- **Smart Matching Algorithm** - Automatically suggests potential matches between found and lost items based on category, location, timing, and description keywords
- **Search & Filter** - Instantly search lost and found items by category and location
- **Analytics Dashboard** - Visual statistics on recovery patterns, common loss locations, and recovery rates by category
- **CRUD Operations** - Full create, read, update, delete support on both collections

Technology Stack

- **Backend:** Node.js, Express.js (ES6 modules)
- **Database:** MongoDB Atlas (native driver, no Mongoose)
- **Frontend:** Vanilla JavaScript (ES6 modules), HTML5, CSS3
- **Tools:** ESLint, Prettier, Nodemon

Goals

- Reduce time between "item lost" and "item recovered"
- Provide a centralized platform replacing scattered group chats and emails

- Surface recovery patterns to help students know where to search
- Connect finders and losers proactively through smart matching

---

2. User Personas

Persona 1: Will - The Panicked Student

Age: 20 | **Year:** Sophomore

Goals:

- Quickly check if someone already found and reported her lost laptop
- Submit a lost item report with all relevant details
- Get notified of potential matches

Pain Points:

- Has to check multiple group chats and physical boxes across campus
- No centralized place to search for found items
- Wastes time retracing steps without knowing if item was already found

How CampusRecover Helps:

- ✅ Instant search across all found item reports
- ✅ Smart matching alerts her to likely matches
- ✅ Single platform replacing scattered group chats

Persona 2: Mikayla - The Good Samaritan

Age: 22 | **Year:** Senior

Goals:

- Return a found wallet to its owner quickly
- Avoid posting on multiple group chats
- See if anyone reported losing the item she found

Pain Points:

- No easy way to find the owner without posting everywhere
- Doesn't know where to hand in found items
- Worried the item won't reach the right person

How CampusRecover Helps:

- ✅ Submit one found item report and let the system find matches
- ✅ Smart algorithm connects her report to the likely owner
- ✅ Direct contact info exchange between finder and owner

Persona 3: Officer Martinez - Campus Security

Age: 38 | **Experience:** 10 years

Goals:

- Monitor high-value item reports across campus
- Identify patterns in where items are commonly lost
- Assist in recovery efforts efficiently

Pain Points:

- No visibility into what students are losing/finding
- Cannot identify hotspot locations without data
- Reactive rather than proactive approach to lost items

How CampusRecover Helps:

- ✅ Analytics dashboard shows common loss locations
- ✅ Recovery rate data identifies problem areas
- ✅ Centralized platform gives campus-wide visibility

---

3. User Stories

Preshit Ravindra Pimple - Lost Items & Analytics

Story 1: Search Found Items As a student who just lost an item, I want to search existing found items by category, location, and date, so that I can quickly check if someone already reported finding it.

- Search returns relevant results within seconds
- Filters work by category dropdown and location text
- Results sorted by most recent first

Story 2: Submit Lost Item Report As a student, I want to submit a detailed lost item report with description, last seen location, date/time, and contact info, so that others who find it can reach me.

- Form validates all required fields
- Date cannot be set to the future
- Item appears in list immediately after submission

Story 3: Edit or Delete Lost Reports As a student, I want to edit or delete my lost item reports, so that I can update details or remove recovered items.

- Edit modal pre-fills existing data
- Delete requires confirmation
- Changes reflect immediately in the list

Story 4: Mark Item as Recovered As a student, I want to mark my lost item as recovered, so that the platform reflects successful recoveries.

- One-click status change to "recovered"
- Recovered items sorted below active items
- Status badge updates visually

Story 5: View Analytics As a platform user, I want to view analytics on recovery patterns including common loss locations, recovery success rates by item type, and most frequently lost items, so that I understand campus-wide patterns.

- Dashboard loads all stats in parallel
- Visual bars show location frequency
- Recovery rate table shows per-category breakdown

Gurudatt Pramod Gaonkar - Found Items & Matching

Story 1: Report a Found Item As a student who found an item, I want to report found items with description, location found, current holding location, date/time, and contact info, so that I can help return items to their owners.

- Form captures both where item was found and where it is now
- Date cannot be set to the future
- Item appears in list immediately after submission

Story 2: Edit or Delete Found Reports As a student, I want to edit or delete my found item reports, so that I can update pickup locations or remove claimed items.

- Edit modal pre-fills all existing fields

- Delete requires confirmation prompt
- Updates reflect immediately

Story 3: Mark Item as Claimed As a student, I want to mark found items as claimed, so that others know they have been returned to the owner.

- One-click status change to "claimed"
- Claimed items sorted below unclaimed items
- Status badge updates visually

Story 4: Smart Match Suggestions (Finder View) As a student who found an item, I want the system to automatically suggest potential matches from recent lost reports based on description, category, location proximity, and timing, so that I can proactively contact the likely owner.

- Matches scored out of 100 based on 4 criteria
- Top 5 matches shown with confidence percentage
- Human-readable reasons explain why each item matches

Story 5: Smart Match Suggestions (Loser View) As a student searching for my lost item, I want to see potential matches when viewing found items, so that I can quickly identify items that might be mine.

- Match score color-coded: green (high), yellow (medium), gray (low)
- Owner contact info shown directly on match card
- "No matches" state shown clearly when none found

---

4. Matching Algorithm Design

Concept

When a user clicks "🎯 Matches" on a found item, the system queries all **active** lost item reports and scores each one against the found item across four dimensions.

Scoring Breakdown (Total: 100 points)

Criteria	Points	Method
Same category	40	Exact string match
Similar location	30	Case-insensitive partial string match

Date proximity	20	Found within 7 days of when item was lost
Keyword similarity	10	Jaccard similarity on description tokens

Match Confidence Levels

- 🟢 **High** (≥70%) - Strong match, very likely the same item
- 🟡 **Medium** (≥40%) - Possible match, worth investigating
- ⚫ **Low** (<40%) - Weak match, shown as last resort

Example

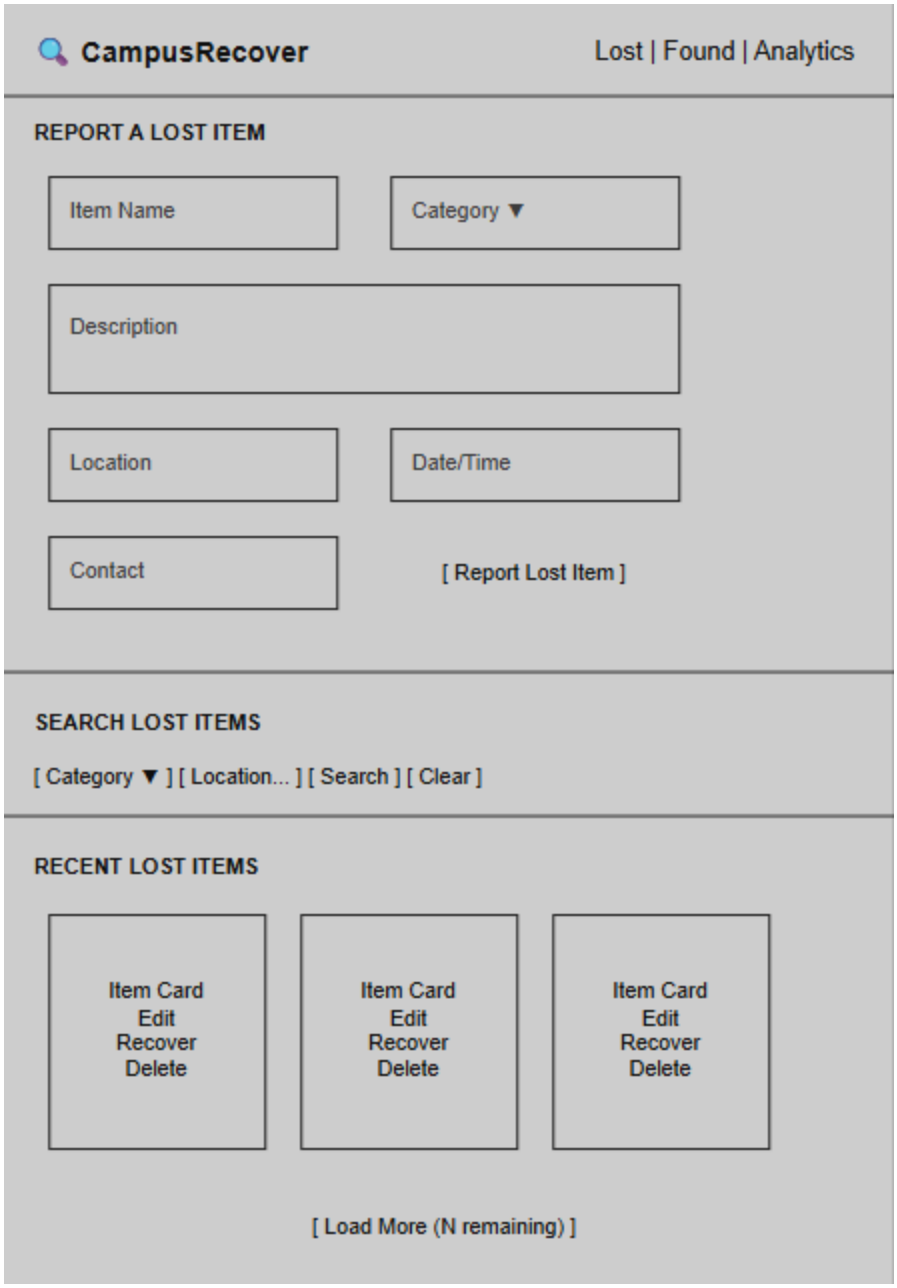
A found **Electronics** item at **Snell Library** on **Day 5** with description "black MacBook charger" will score highly against a lost **Electronics** item at **Snell Library 3rd floor** from **Day 3** with description "MacBook charger, black cable":

- Category match: +40
- Location match: +30
- Date within 7 days: +20
- Keyword overlap ("macbook", "charger", "black"): +8
- **Total: 98% match**

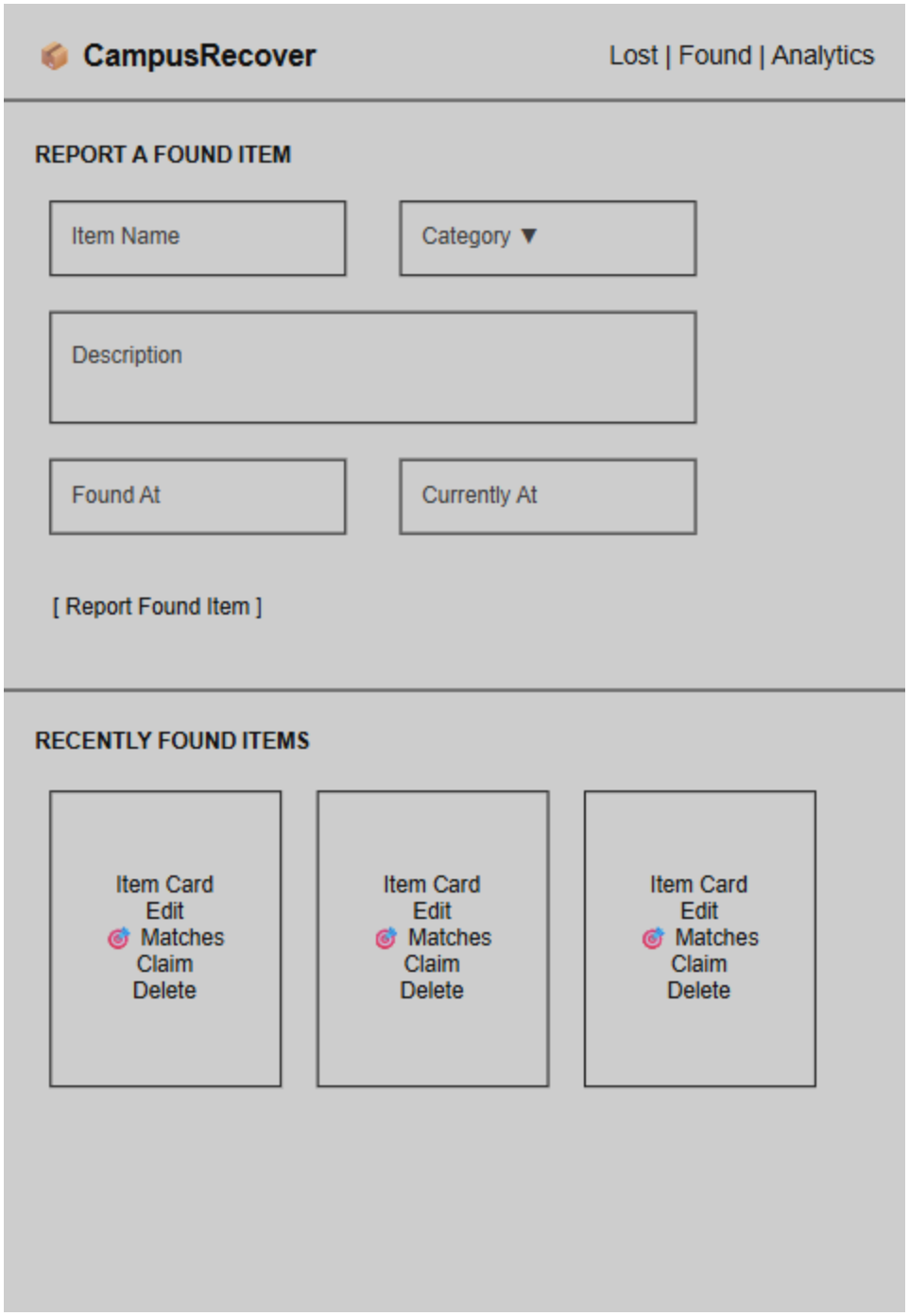
---

5. Wireframes

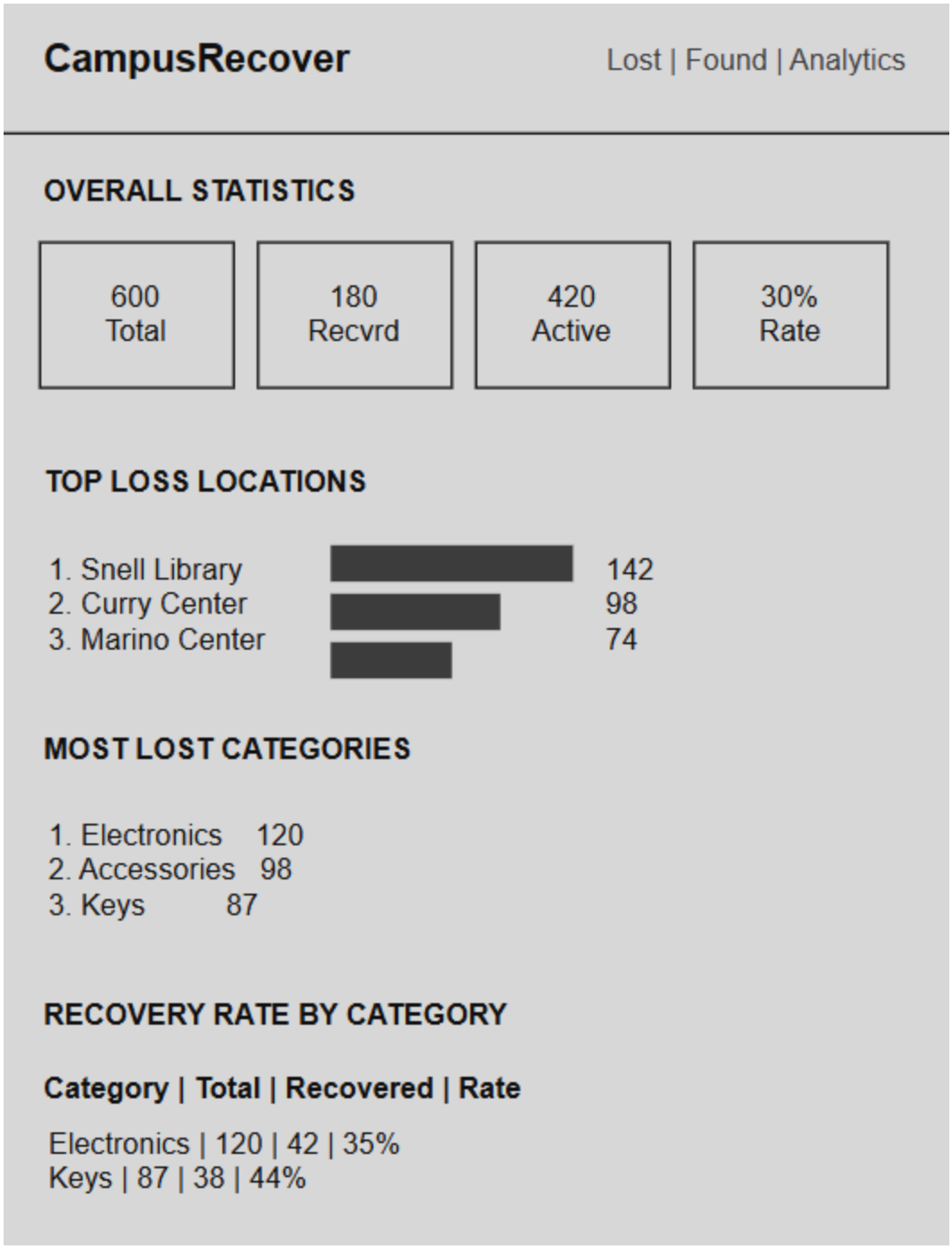
Page 1: Lost Items Page (index.html)



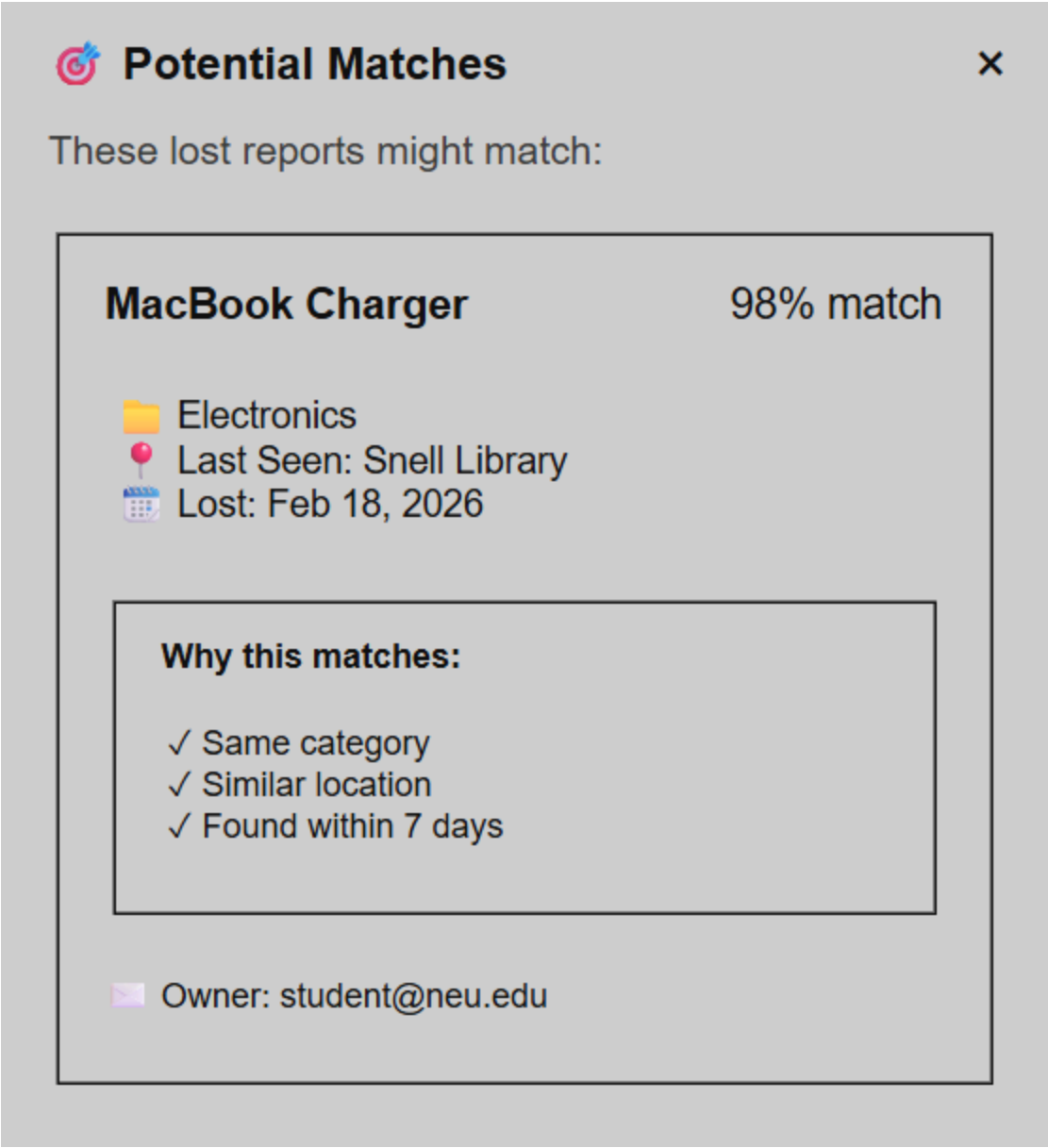
Page 2: Found Items Page (found.html)



Page 3: Analytics Dashboard (analytics.html)



Matching Modal Wireframe



---

6. Database Schema

lost_items Collection

```
{
  _id: ObjectId,
  itemName: String,
  category: String,     // Electronics | Accessories | Clothing | Books | IDs | Keys | Other
  description: String,
  location: String,     // Last seen location
```

```
  dateTime: Date,       // When item was lost
  contactInfo: String,  // Owner's email
  status: String,       // "active" | "recovered"
  createdAt: Date,
  updatedAt: Date
}
```

found_items Collection

```
{
  _id: ObjectId,
  itemName: String,
  category: String,         // Electronics | Accessories | Clothing | Books | IDs | Keys | Other
  description: String,
  locationFound: String,    // Where item was found
  currentLocation: String,  // Where item is being held
  dateTime: Date,           // When item was found
  contactInfo: String,      // Finder's email
  status: String,           // "unclaimed" | "claimed"
  createdAt: Date,
  updatedAt: Date
}
```