

FindIt: Lost & Found Management System

Kayleb Bassas, Angel Collado, David Hernandez, Deijen Severino, Alejandro Hernandez · Instructor: Prof. Masoud Sadjadi

Problem

- Lost items on campus are often reported through scattered methods, making them difficult to locate.
- Students and staff may not know where to search for lost property or report items they have found.
- Existing methods such as social media, group chats, and physical lost-and-found locations are inefficient and unorganized.
- There is no centralized platform for searching and managing lost-and-found listings.
- Users need an easier way to filter listings by keywords, category, status, location, and date.
- A centralized system could improve communication and increase the likelihood of successfully recovering lost items.

Stakeholders & Requirements

Stakeholders

- Students and campus community: Primary users who report, search for, and recover lost items.
- Faculty and staff: Users who may report or search for lost and found property.
- Campus administrators: Responsible for monitoring listings, managing inappropriate content, and handling ownership disputes.
- Project team: Designs, develops, tests, and maintains the application.

Functional Requirements

- Users can create accounts and securely log in.
- Users can create, edit, search, filter, and manage lost-and-found listings.
- Listings include item name, category, description, location, date, and images.
- Users can communicate with other users regarding posted items.
- Users can mark items as found or recovered.
- Users can manage their profiles and listings.

Non-Functional Requirements

- Secure authentication using JWT and password hashing.
- Responsive and user-friendly interface.
- Reliable database storage and retrieval.
- Modular and maintainable system architecture.
- Fast search and responsive application performance.

Our Solution

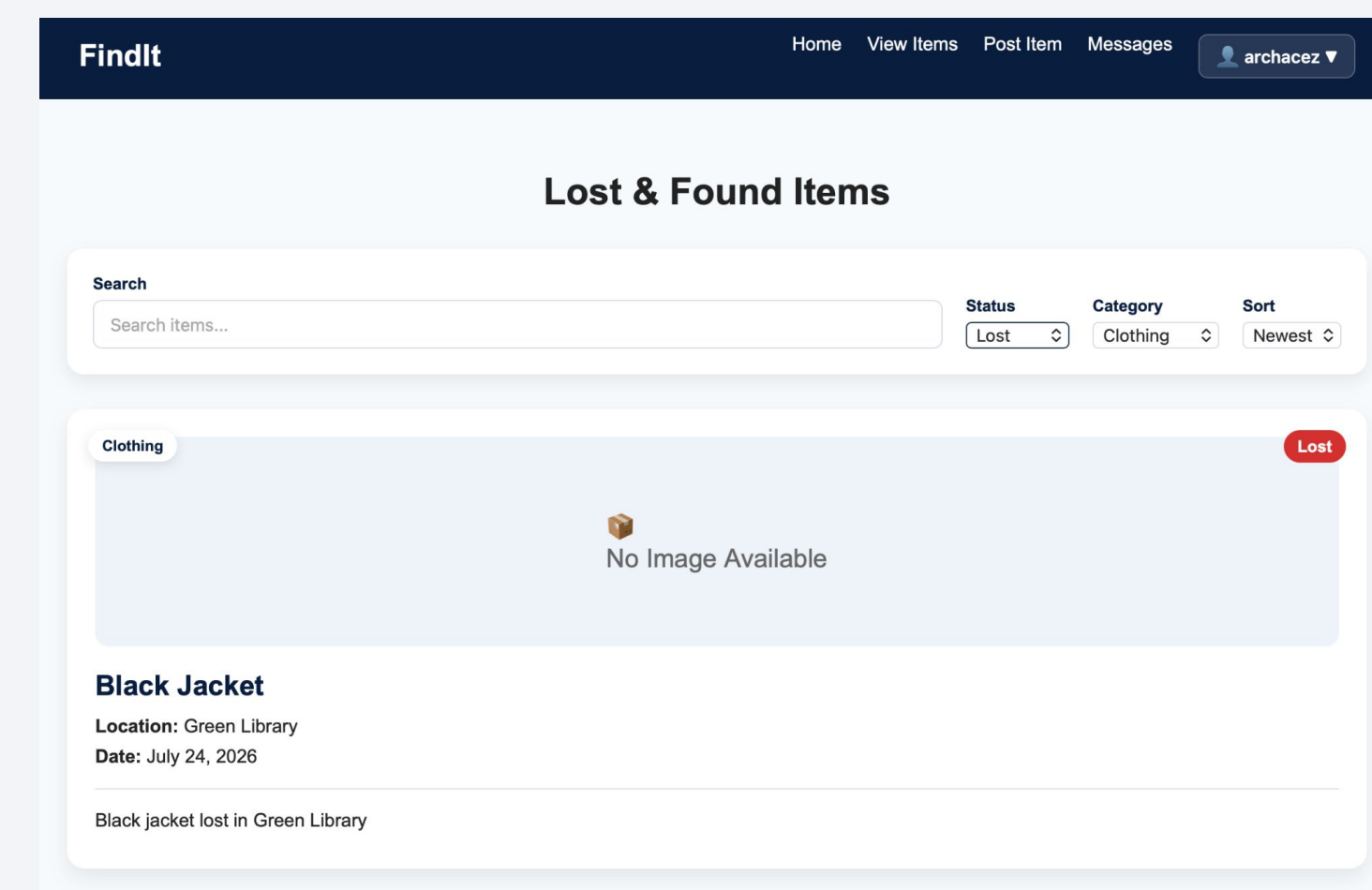
Overview

FindIt is a full-stack web application that provides a centralized platform for reporting, searching, and recovering lost and found items. Users can create and manage item listings, search existing posts, and communicate directly with one another through an integrated messaging system. The application was developed using an Agile Scrum process over five development sprints.

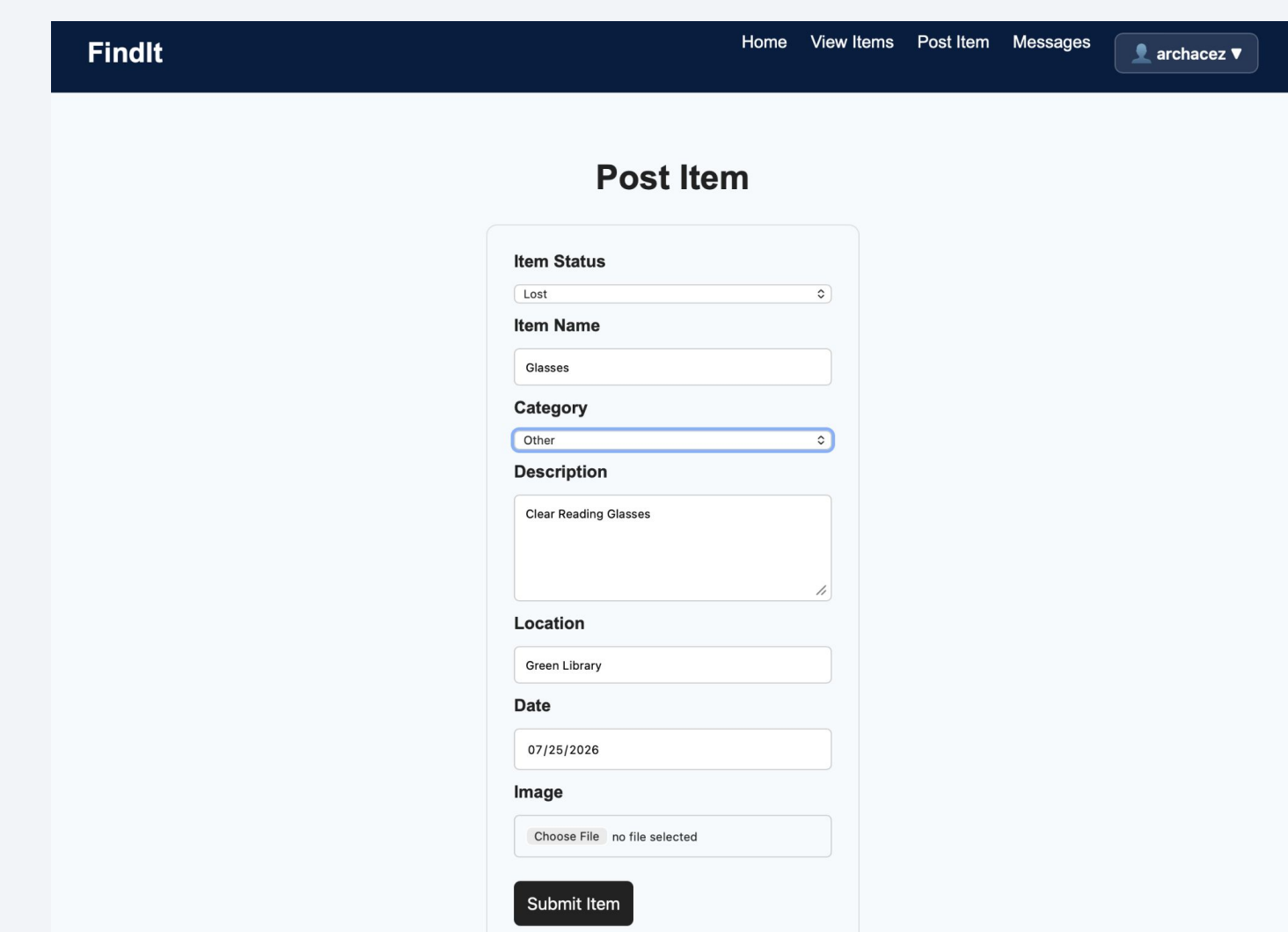
Design Decisions

- React frontend with an Express.js backend to separate the user interface from application logic, improving organization and maintainability.
- MongoDB Atlas to store users, item listings, and messages using a flexible document-based database.
- JWT authentication with bcrypt to provide secure user authentication and protect user accounts and application features.

Demo & Screenshots



"Users can search and filter listings by keyword, status, category and sort to quickly find a match."



"Reporting a lost item takes a status, name, category, description, location, Date and photo."

Architecture

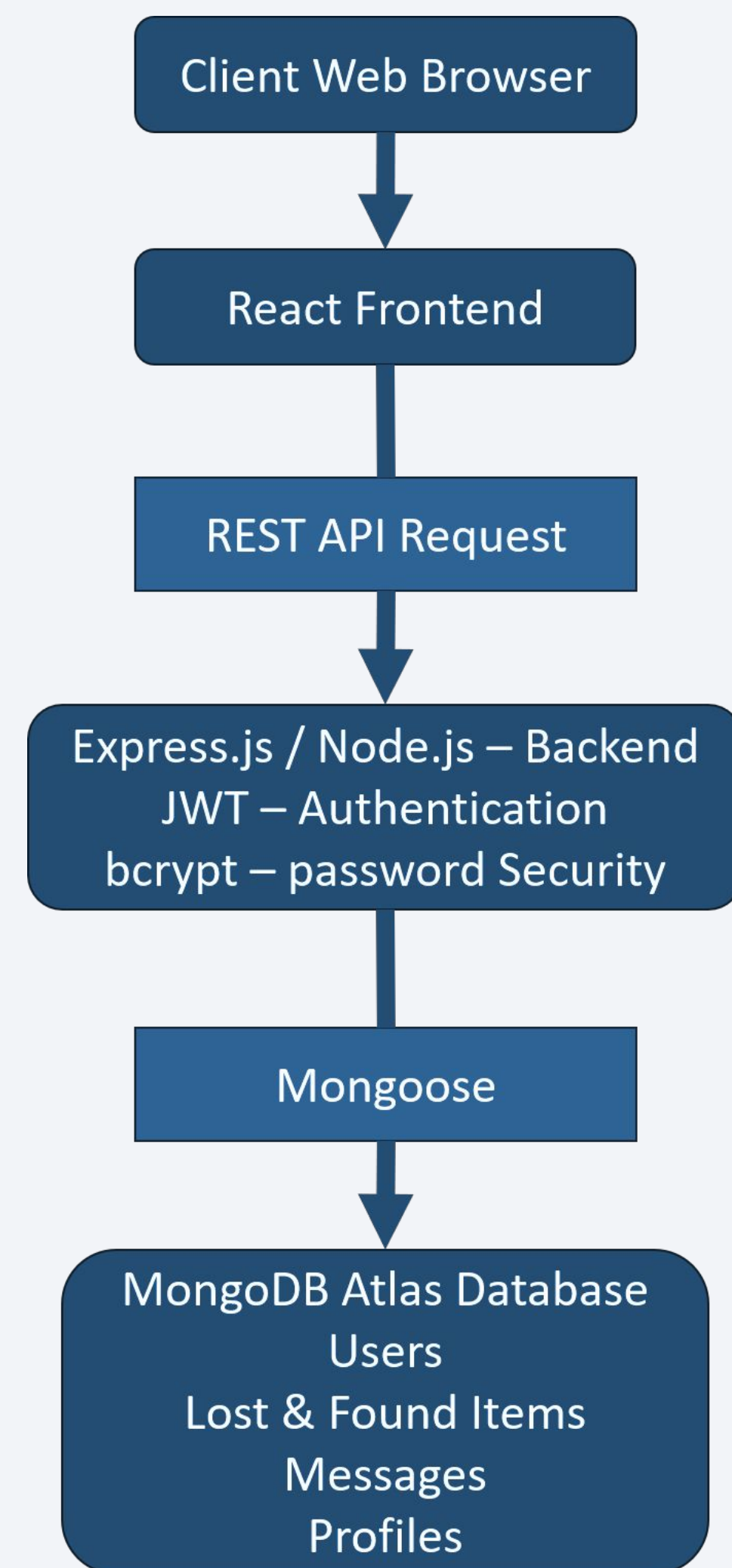


Figure 1. FindIt System Architecture

Engineering Process (SO6)

- Agile Scrum SDLC
 - Five development sprints
 - Sprint Planning, Daily Scrums, Reviews & Retrospectives
 - Continuous backlog refinement
- Design Decisions
 - React frontend
 - Express.js REST API backend
 - MongoDB Atlas database
 - JWT authentication & bcrypt password hashing
- Implementation
 - Modular client-server architecture
 - CRUD operations for lost & found items
 - User messaging and profile management
- Testing & Verification
 - Manual functional testing
 - Postman API testing
 - Frontend/backend integration testing
 - System walkthroughs and usability testing

What's Next

- Enhance application security with Google authentication, email verification, and password recovery.
- Develop an administrator dashboard for reviewing reports, managing listings, and handling recovered items more efficiently.
- Improve the user experience through continued interface refinement and additional usability enhancements.
- Explore AI-assisted item matching to automatically identify potential matches between lost and found items.
- Add real-time location tagging and push/email notifications to alert users when a potential match is found.
- Continue expanding functionality based on user feedback and future campus needs.

Acknowledgments

We thank the FIU Knight Foundation School of Computing and Information Sciences, the Capstone Program, Prof. Masoud Sadjadi, and everyone who provided guidance and feedback throughout the development of this project.