CU Lost & Found Web-application

Development team: Andrew Settergren, Alex Sorensen, Ian Wong, Kelsey Valencia, and Nikkii Fox

Purpose

The CU Boulder Lost & Found web-application was designed to become a new secure resource for students and faculty who have lost or found items on campus.

Tools Used

	Jira Software	Product Management	★★★
	GitHub	VCS Repository	★★★★
	MySQL	Database	★★★★★
	Embedded JavaScript	Templating Engine	★★★★★
	Visual Studio Code	IDE	★★★★★
	Heroku	Deployment environment	★★★★

Product Management: Jira Software

★ ★ ★

Backlog

Share ···

Only My Issues   Recently Updated

EPICS	Create epic ✕

All issues

> Basic-Homescreen

> Registration-Page

> Manage Listings

> Lost & Found Office Functions
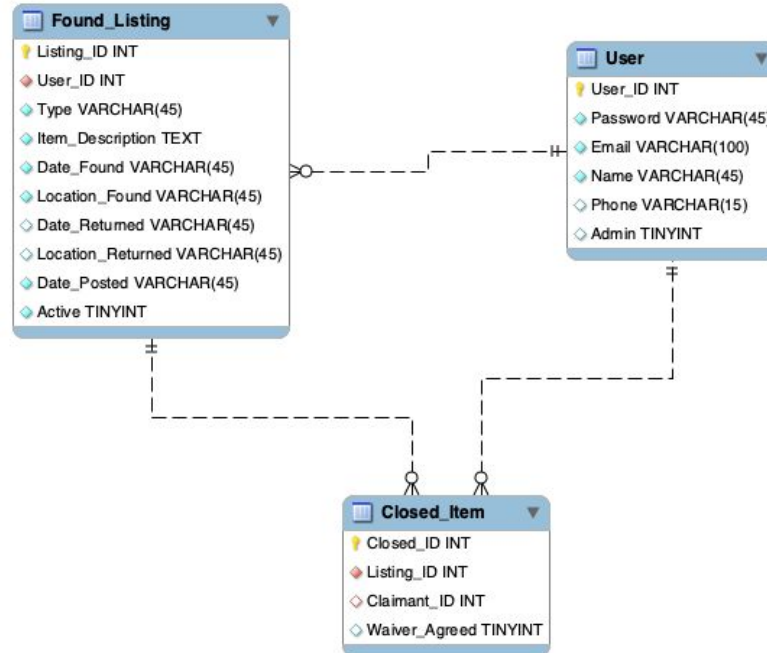
> Notification/Messaging System

> Create Server

Issues without epics

Create Server	Create Server	AS	S17-28	↑	-
Link front-end to server	Create Server	AS	S17-29	↑	-
Link back-end to server	Create Server		S17-30	↑	-
Edit a listing for a lost or found item	Manage Listings		S17-10	↑	-
Remove a listing for a lost or found item.	Manage Listings		S17-6	↑	-
Interface for Sign-In/Create Account page (Identikey?)	Registration-Page	AS	S17-4	↑	-
MySQL Database to store user/listing data	Registration-Page		S17-11	↑	-
Add drop-down menu linking to other pages within the application	Registration-Page	AS	S17-12	↑	-
Add menu linking to all areas of the application	Basic-Homescreen	AS	S17-14	↑	-
Interface (Layout, banners, text)	Basic-Homescreen	AS	S17-13	↑	-
Interface/layout of search page and results	Manage Listings				
Search for items based on type, color, location, and/or date found	Manage Listings				

Quickstart

Database: MySQL

★ ★ ★ ★ ★

VCS Repository: GitHub

⭐ ⭐ ⭐ ⭐

<%= EJS %> Template Engine: Embedded JavaScript

⭐ ⭐ ⭐ ⭐ ⭐

```
<% if(db_data.length){ for(var i = 0;i < db_data.length;i++) { %>
<div class="card" id="item<%=(i+1)%>">
    <div class="row">
    <div class="col-md-6">
    <img class="card-img-top" src="../resources/images/placeholder.svg" alt="Card image cap" style="margin-left:auto; mar
    padding-bottom:-12px; padding-left:10%; width:85%; height:85%;">
    </div>
    <div class="col-md-6">
      <div style="margin-top:5%; margin-right:2%;">
    <h5 class="card-title"> <%= db_data[i].Type%> </h5>
    <p class="card-text">Description:  <%= db_data[i].Item_Description%> </p>
    <p class="card-text">Date Found:  <%= db_data[i].Date_Found%> </p>
    <p class="card-text">Location Found: <%= db_data[i].Location_Found%> </p>
    <p class="card-text">Location Returned:  <%= db_data[i].Location_Returned%> </p>
    <p class="card-text">Date Returned: <%= db_data[i].Date_Returned%> </p>
    <p class="card-text"><small class="text-muted">Posted  <%= db_data[i].Date_Posted%> </small></p>
</div>
    </div>
    </div>
    <br>
    </div>
    <% } %>
    <% }; %>
```

Framework: NodeJS

★★★★★

CU_Lost_Found > JS server.js > ⬡ app.post('/post_item') callback

```js
1    /////MODULES//////
2    var http = require('http');
3    var express = require('express'); //Ensure our express framework has been added
4    var cookieParser = require('cookie-parser');
5    var session = require('express-session');
6    var app = express();
7    var bodyParser = require('body-parser'); //Ensure our body-parser tool has been added
8    app.use(bodyParser.json());                  // support json encoded bodies
9    app.use(bodyParser.urlencoded({ extended: false })); // support encoded bodies
10   app.use(bodyParser.text());
11   app.use(cookieParser());
12   app.use(session({secret: "shhhhh", saveUninitialized: true,resave: true}));
13   //////////////////
14
```

IDE: Visual Studio Code
★★★★★

Deployment Environment: Heroku

★ ★ ★ ★

User Acceptance Testing

- We used our app as actual users in order to make sure it could handle the required tasks in real-world scenarios.
- We tested all aspects of our app. Some important areas we tested are as follows:
  - Logging in as an existing user + registering as a new user
  - Posting a new item
  - Searching for items
  - Editing an item + changing active status
  - Editing account information

Methodologies

Agile Process

- Using ongoing sprints of project planning and execution, we were able to continuously adapt and mature our plan, scope, and design throughout the project.
- Biweekly meetings with stakeholders
- ★ ★ ★ ★

Peer Code Reviews

- Throughout the semester, we held team meetings twice a week to discuss any setbacks and conduct peer code reviews.
- ★ ★ ★ ★ ★

Challenges

- Collaborating on the project simultaneously through GitHub
- Hosting the web application on Heroku and familiarizing ourselves with the ClearDB environment (for MySQL)
- Working closely alongside stakeholders from CU's Office of Information Technology to understand the expectations and requirements needed in order to integrate our web application into Buff Portal

Demo

Questions/Comments?