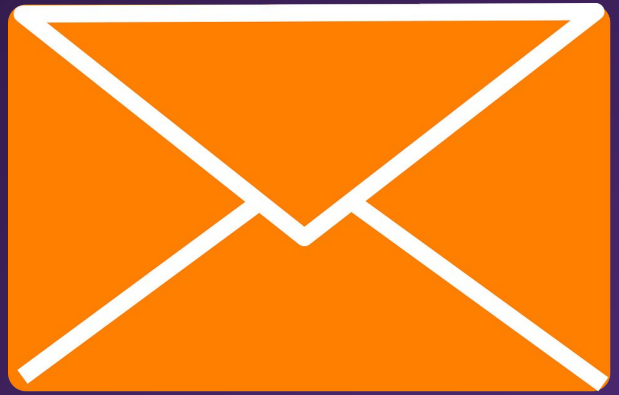
Easy Rec

An escrow file server for letters of recommendation



Developed By Team Easy Rec
Members: Henry Eigen, Jovan Hernandez, Ria Patel, & Ross Ketron

INTRODUCTION

Current Problem:

- A job candidate needs a letter of recommendation for each position
- A candidate can't manage his/her own letter since they are not allowed to read the contents

Current Solution:

- A candidate asks the recommender to send the letter to each position applied for

INTRODUCTION

Referrer Problems:

- Tracking written letters
- Have to write letter for each applicant for each company
- Time Consuming

Applicant Problems:

- Have to rely on recommender to resend letter for each company
- Recommender can forget or miss deadline
- No control over letters

SOLUTION

EasyRec solves the problem of managing letters applicants can't read by providing an escrow process which handles file references without file access

Easy Rec:

- Upload letter once, send letter forever
- Easily manage and track uploaded letters
- Sends letters easily to hiring agents from organizations all in the same place

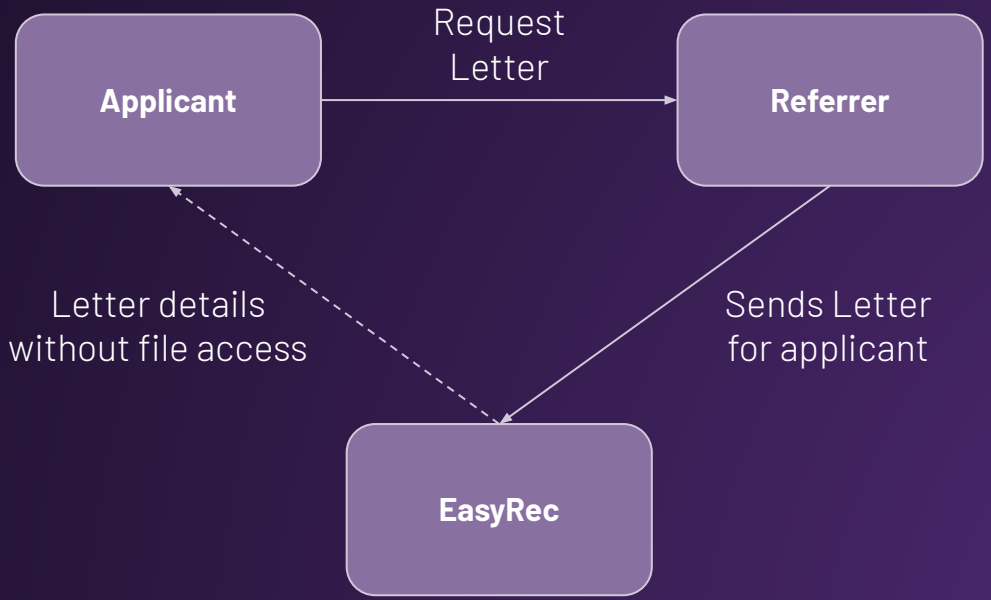
CUSTOMER VALUES

What we accomplish for our users

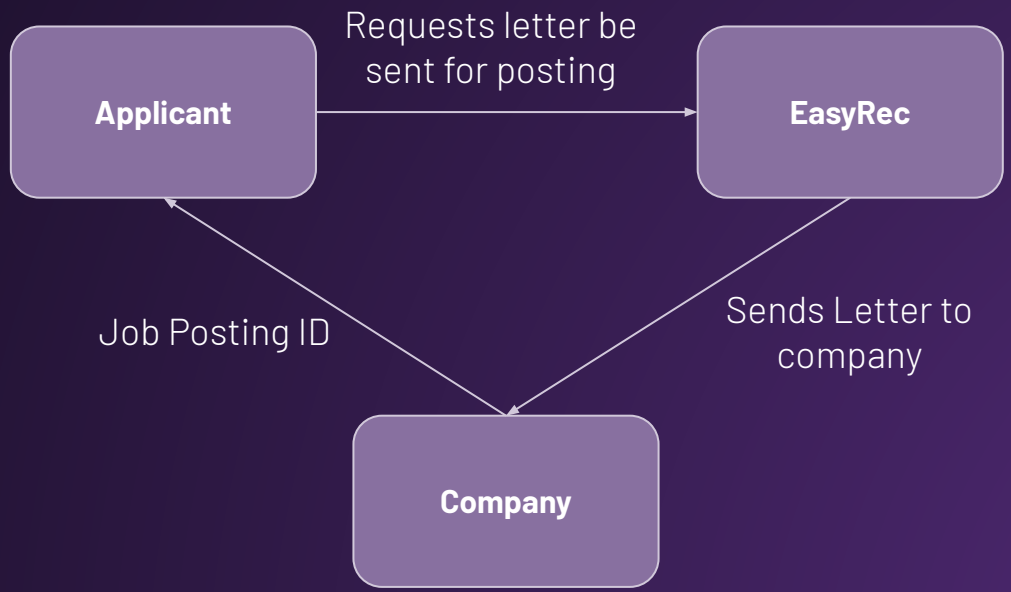
CUSTOMER VALUES

- Primary users include current students and early career professionals
- Applicant can track when the rec letters have been sent
- Referrers only have to upload one rec letter
- Hiring agents can organize and manage campaigns to receive letters

CUSTOMER VALUES (LETTER UPLOAD)



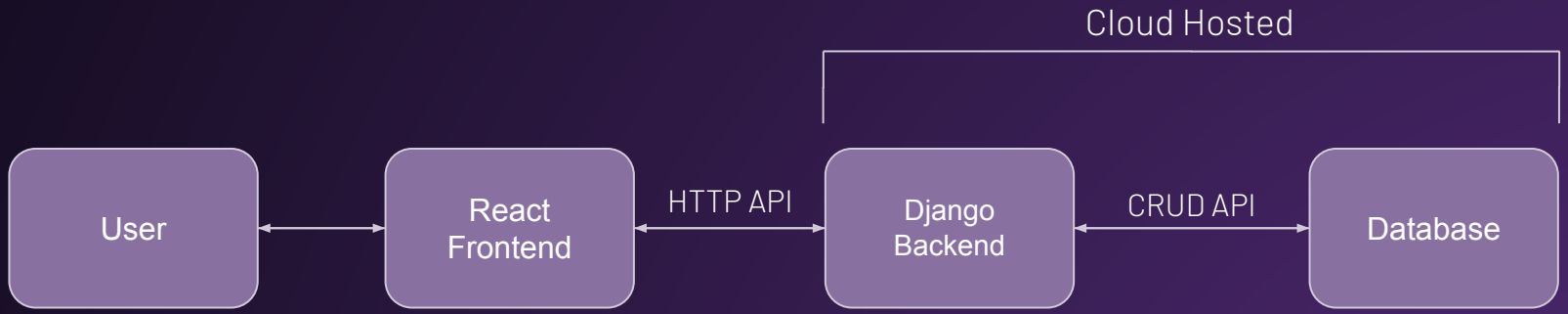
CUSTOMER VALUES (LETTER POSTING)



TECHNOLOGY

How we implemented Easy Rec

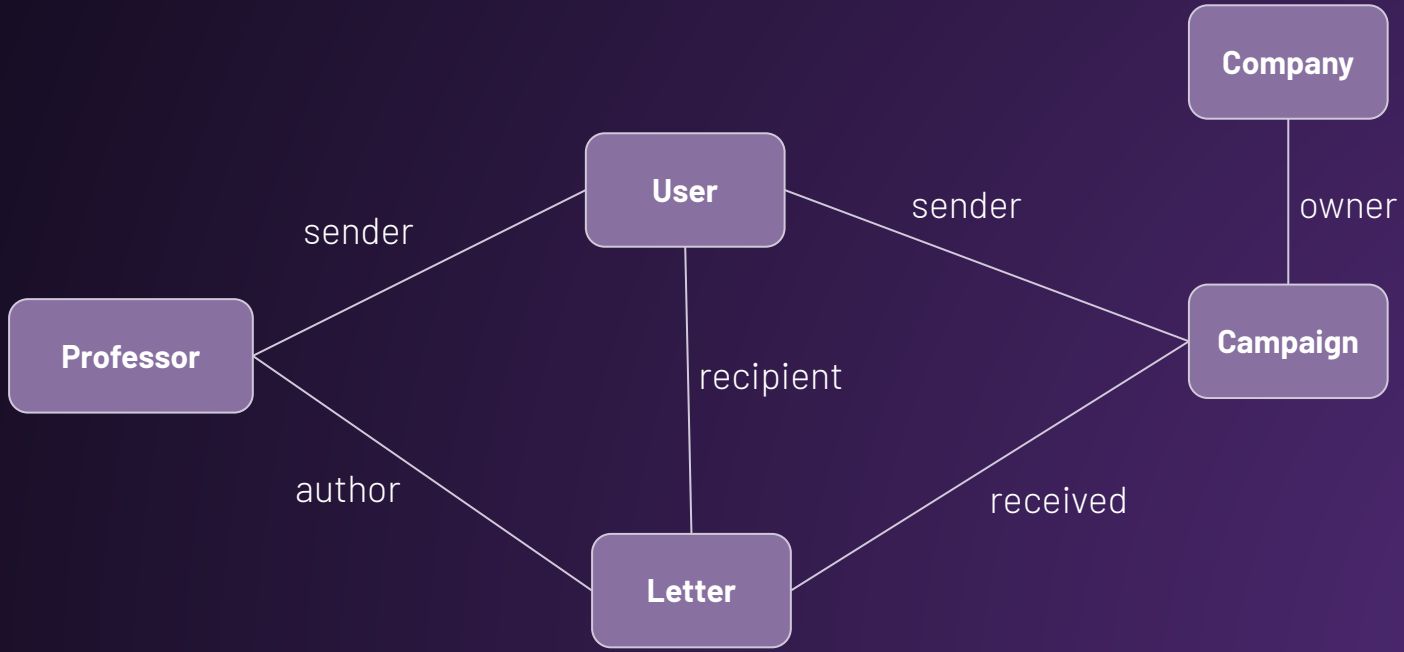
TECHNOLOGY



Changes include:

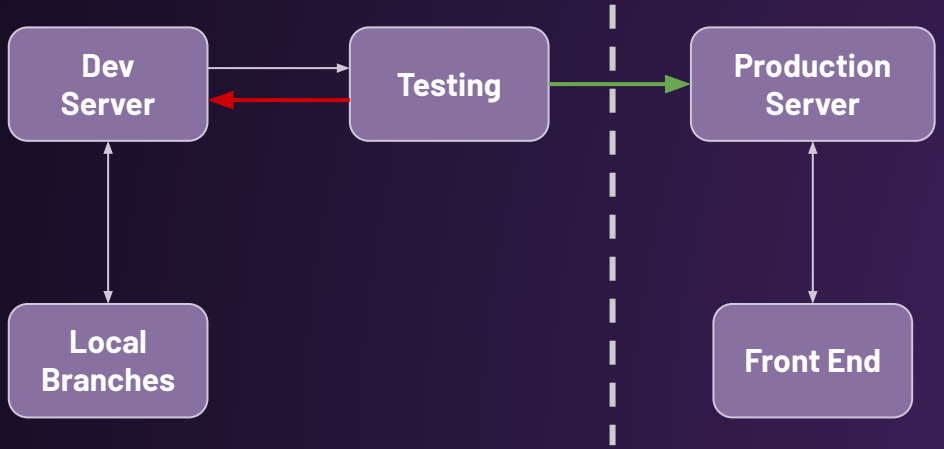
- Deployed backend to the cloud separately from the frontend purely to ease development.
- Stored everything in SQL database instead of a file server.
- Database schemas were adjusted as we progressed.
- Multiple user types were grouped under one struct.

Tangled Relationships



API Continuity

- Tests for each API
- Ensures non breaking commits
- Prevents backend dev changes from interrupting frontend dev




```
Testing User Registration:
100%|████████████████████████████████| 30/30 [00:07<00:00,  4.16it/s]
Succesfully registered: ✅ (30/30)

Testing User Login:
100%|████████████████████████████████| 30/30 [00:06<00:00,  4.33it/s]
Succesfully logged in: ✅ (30/30)

Testing Letter Upload:
100%|████████████████████████████████| 30/30 [00:05<00:00,  5.70it/s]
Succesfully uploaded: ❌ (28/30)

Testing Retrieve Letter:
100%|████████████████████████████████| 30/30 [00:05<00:00,  5.94it/s]
Succesfully retrieved: ✅ (30/30)

Testing campaign creation:
100%|████████████████████████████████| 30/30 [00:05<00:00,  5.92it/s]
Succesfully created: ✅ (30/30)

Testing Campaign Lookup:
100%|████████████████████████████████| 30/30 [00:05<00:00,  5.98it/s]
Succesfully searched: ✅ (30/30)

Testing Send Letter:
100%|████████████████████████████████| 30/30 [00:05<00:00,  5.93it/s]
Succesfully sent: ❌ (24/30)

Deleting Dummy Users:
100%|████████████████████████████████| 30/30 [00:05<00:00,  5.62it/s]
Succesfully deleted: ✅ (30/30)

Verifying Letters Automated Cleanup:
100%|████████████████████████████████| 30/30 [00:05<00:00,  5.17it/s]
Succesfully verified: ✅ (30/30)

Verifying Campaigns Automated Cleanup:
 47%|███████████████                  | 14/30 [00:02<00:03,  5.15it/s]
```

DEVELOPMENT

Did go well:

- Mapped out ahead of time
- Development pipeline worked well
- Separation of frontend and backend

Issues:

- Cloud deployment issues

TEAM

An introduction to our members

TEAM

Back-end Development:

- Henry Eigen
  - Roles: Wrote backend API's, designed SQL schema, managed cloud hosting of backend
  - Skills: Python, Django
- Ria Patel
  - Roles: Project Manager, assisted with back-end, assisted with mock ups, status reports
  - Skills: Python, Javascript

TEAM

Front-end Development:

- Ross Ketron
  - Roles: collaborated on React components, assisted with mockups, front-end testing
  - Skills: HTML, CSS, Javascript, React, MaterialUI
- Jovan Hernandez
  - Roles: collaborated on React components, assisted with mockups, front-end testing
  - Skills: HTML, CSS, Javascript, React, MaterialUI

PROJECT MANAGEMENT

Reaching our goals

PROJECT MANAGEMENT

Time Management:

- Unforeseen challenges meant development schedule was fluid
- Certain features were put off until later
- Others were addressed earlier than planner (e.g. cloud deployment)
- Most unforeseen challenges came in frontend/backend integration

We completed all features for a minimum viable product

PROJECT MANAGEMENT

Future features:

- Separate account types (student vs. professor vs. company)
- API Authentication for security
- Email integration for letter requests
- Permissions for letters (e.g. “Notify me before letter is sent”)
- Human verification for company sign up

DEMO TIME

Video demonstration of our app

https://drive.google.com/file/d/1w5gZ3oCgeSLsjIcYJ4TuM_Ktm1Xo20Vh/view