

MICHIGAN STATE
UNIVERSITY

Project Plan

ARIN Application Launcher

The Capstone Experience

Team Principal AAL

Matt Adomshick

Maegan Johnson

Raaed Khan

Seth Killian

Shiyi Zhao

Department of Computer Science and Engineering
Michigan State University



From Students...
...to Professionals

Spring 2020

Functional Specifications

- Unified interface to view, edit, and create applications
- Manage access to applications
- Single-sign on access to applications

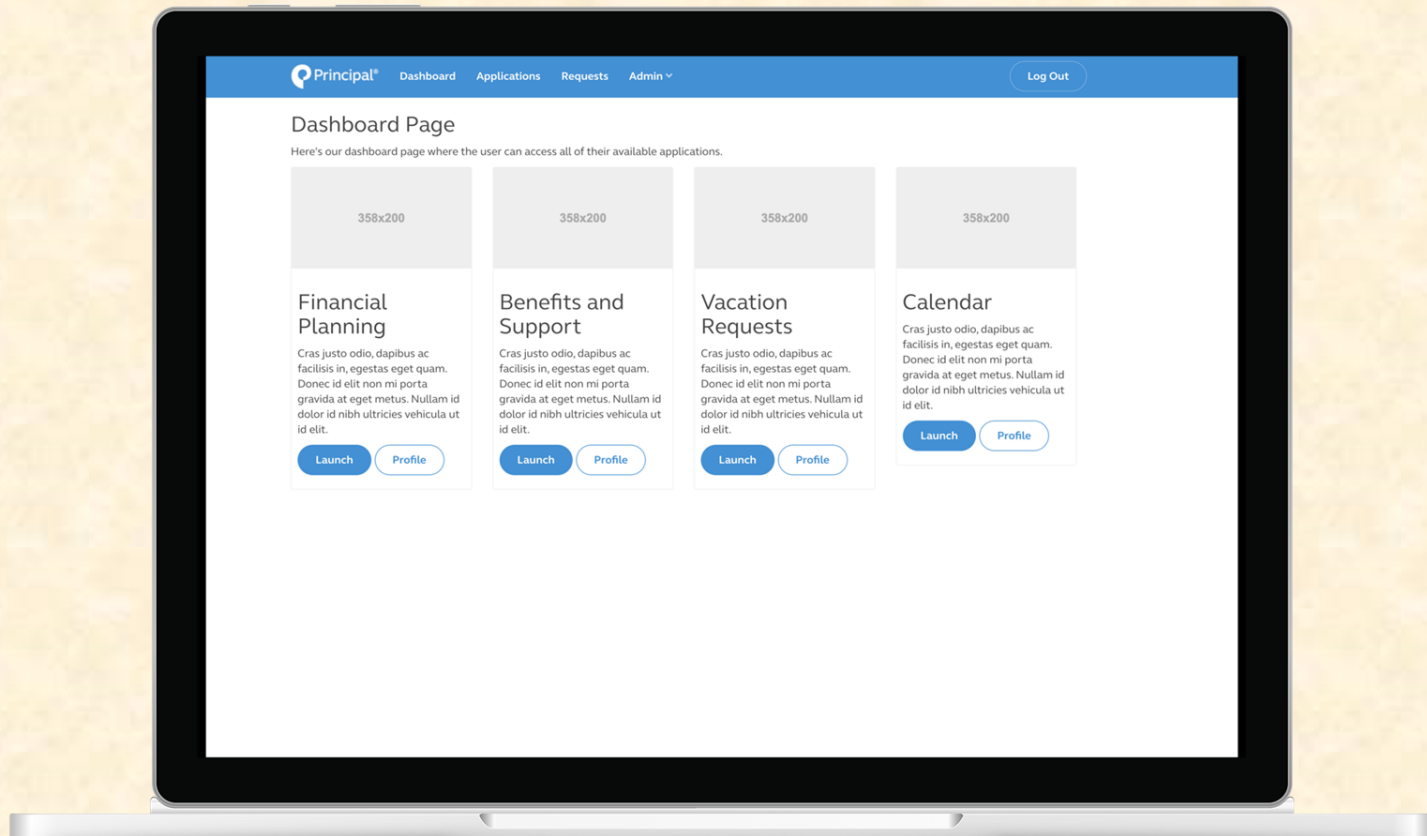


Design Specifications

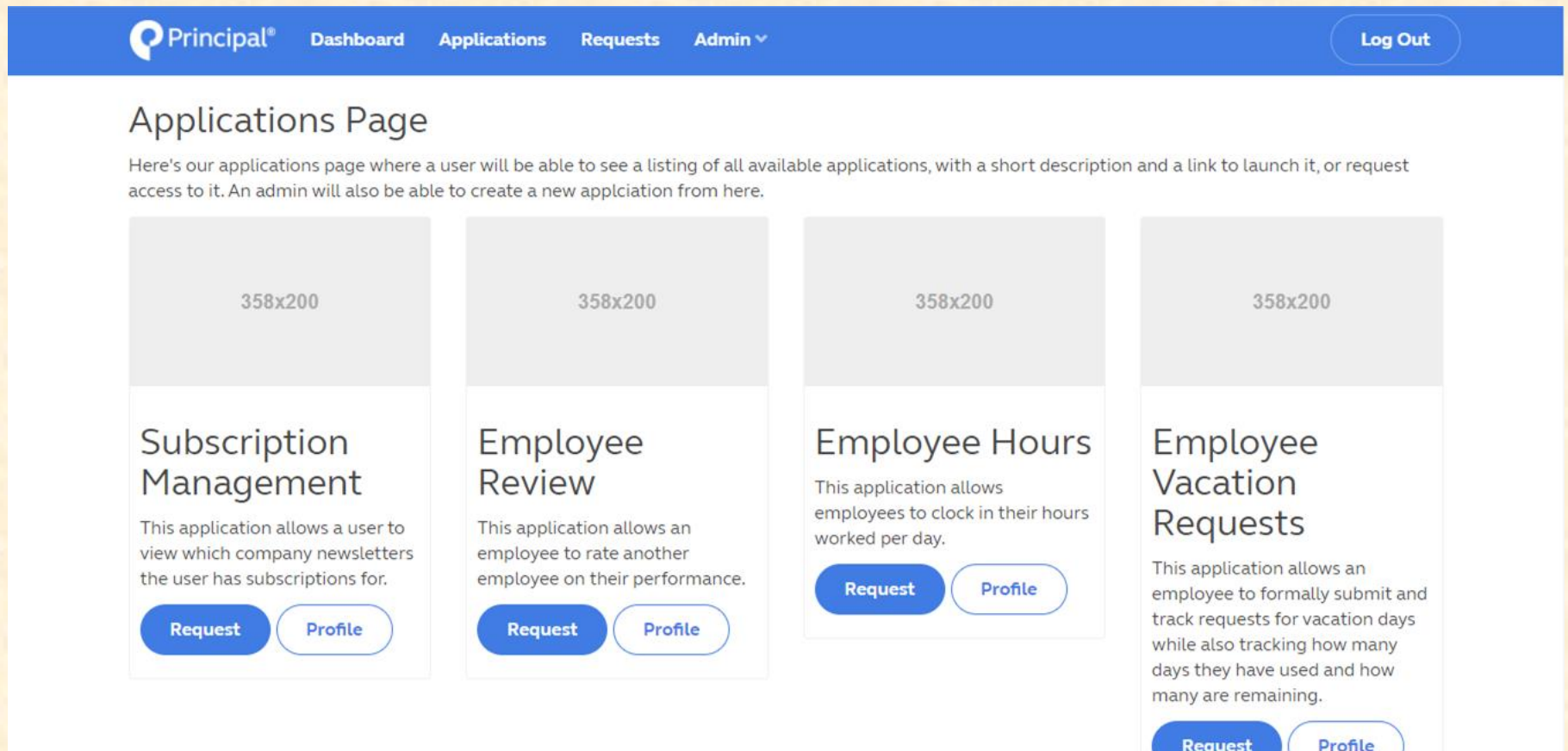
- A navigation bar that uses routing to display every page to the user.
- Display a portfolio of applications that users can access.
- Display tools to administrators that allows them to easily edit permissions and create new applications.




Screen Mockup: User Dashboard



Screen Mockup: Applications



Screen Mockup: Requests

 [Dashboard](#) [Applications](#) [Requests](#) [Admin](#) Log Out

Requests Page

Here's the requests page where a user can see the status of their pending requests and an admin can approve or deny them.

| # | First Name | Last Name | Username |
|---|------------|-----------|----------|
| 1 | Mark | Otto | @mdo |
| 2 | Jacob | Thornton | @fat |
| 3 | Larry | the Bird | @twitter |



Screen Mockup: Administrator Tools

The mockup shows a web interface for creating a new application. It features a blue header with the Principal logo and navigation links. The main content area has a title, a descriptive sentence, and a series of input fields for application details, followed by a submit button.

Principal® Dashboard Applications Requests Admin ▾ Log Out

New Application Page

Here's the page where the admin create a new application.

Application Name:

URL:

Description:

Role1:

Role2:

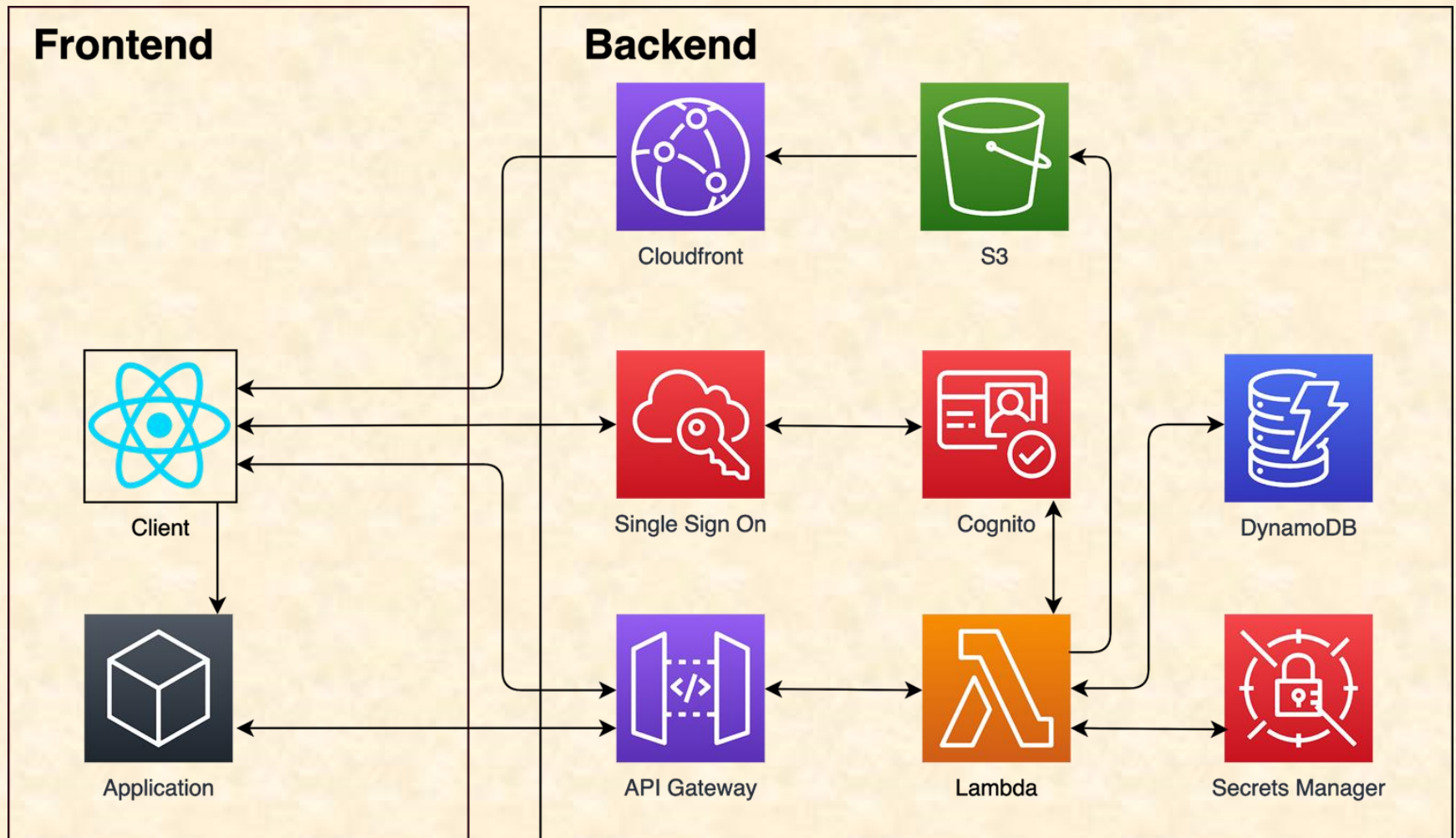
Role3:



Technical Specifications

- Frontend: React and Flask
 - React - JavaScript library for user interface building
 - Flask - Python web app framework for local testing
- Backend: Amazon Web Services
 - Cognito for login/oauth
 - S3 bucket for file storage
 - DynamoDB for databases
 - Amazon API Gateway for API development
 - Lambda for hosting and running code
 - Cloudfront for cloud services

System Architecture



System Components

- Hardware
 - Accessed via internet browser; available on any device that has one
- Software
 - Frontend: React and Flask
 - Backend: Amazon Web Services
 - Cognito, DynamoDB, Amazon API Gateway, Cloudfront, Lambda, S3



Risks

- Serverless Architecture
 - Difficulty: Medium. Importance: Medium.
 - No team member has worked with serverless architecture
 - Online tutorials and sample projects.
- Login with Safety
 - Difficulty: Medium. Importance: High.
 - Login through Cognito and define the correct user pool
 - Draw diagrams to divide groups and use template data to test login
- API Gateway
 - Difficulty: Medium. Importance: High.
 - Hard to define files and functions with different authentication
 - Draw UML diagram to define every input, output and method
- App Module
 - Difficulty: Medium. Importance: Medium.
 - Design of apps' data used for creating, display, and else.
 - Ask client for previous data and comprehensive them with new data



Questions?

?

?

?

?

?

?

?

?

?

