

# ENSE470, Milestone 6

Team : ENSE Master Race  
Jeff Redmond  
Jonathon Wells  
Yilin Ren



# Executive Summary

## Technology Used

- Frontend: Vue.js
- Backend: Node.js + Express
- Database: PostgreSQL
- Hosting: Heroku

Fill in request ticket and submit			Approver granting approval			Analyst vets request		Analyst provides access		User confirms and closes ticket
Create or Sign-in an account	Select software	Send request	Approver receives notification	Approver checks the information provided from the ticket	Approver provides decision for the ticket	View user's information	Progress ticket to next step	If access Is granted send permissions to download softwar	If access is not granted send notification to user	Confirm and close ticket

Release 1

Add sign up form for web application	Add "Software Request" page	Create request ticket	Add notification drawer	Add ticket details page	Add buttons to grant or deny approval	Display fields from user's account	Add option to move ticket to next category	Add button to provide access	Add button to deny access	Display granted key with link to software download
Add sign in form for web application	Add search bar with keyword lookup	Add ticket category: "Waiting for Approval"	Add page to view your assigned tickets	Display user account information	Move ticket to appropriate category		Add categories: "Confirm and Close", "Attention Needed"	On button press, moves the ticket to "Confirm and Close" and send	On deny access, move ticket to "Confirm and Close"	Add button to confirm and close the ticket
	Assign approver based on list for the selected software	Send notification to assigned approver through email and web ap			Add categories: "Approved", "Not Approved"			Add key to user's account	Add user notification/email with reason for denial	

Release 2

Sign up with regular account	Add fuzzy searching		Feature that checks user info with a database to check if it is co	For not approved ticket, approver can add reason/ explanation					For non approval add textbox to add reason	Add ticket comments so user can talk to analyst
------------------------------	---------------------	--	--	---	--	--	--	--	--	---

Sign up with Facebook/Google account	Add search by categories and search filters									
--------------------------------------	---	--	--	--	--	--	--	--	--	--

Finished

Backlogged

Cancelled

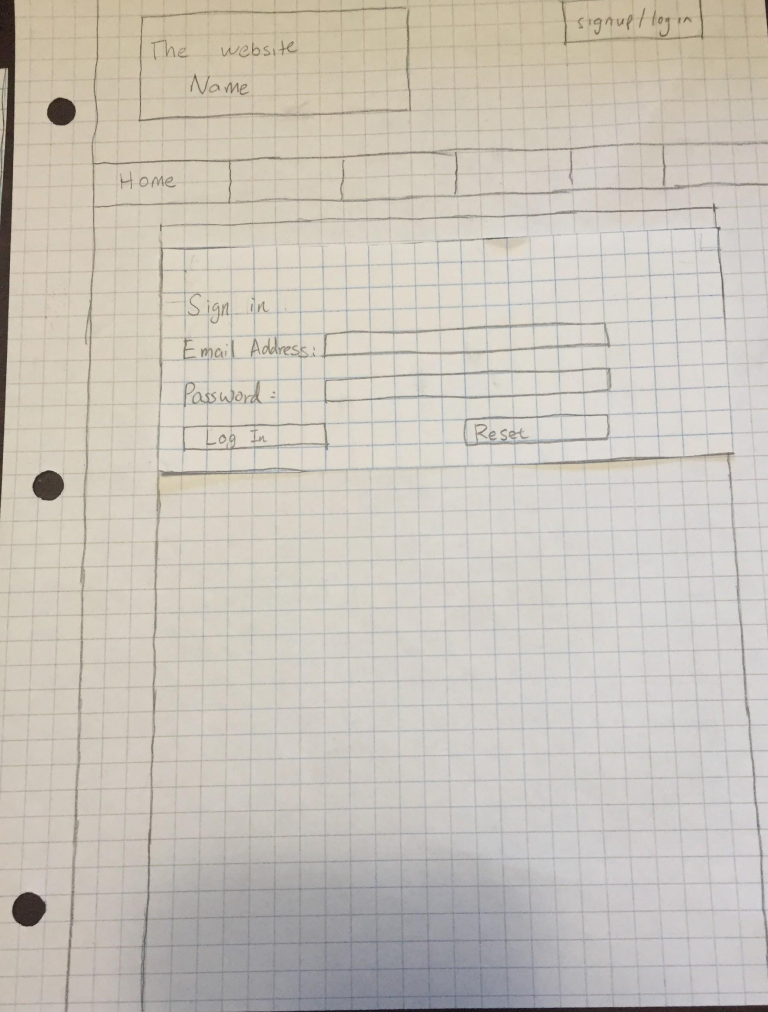


## Welcome

This application is a portal for requesting software. If you are an approver, you can manage the requests for your software here as well.

Please login to continue

[Log in](#)[Sign up](#)



## Log in

[Log in](#)

Dont have an account? [Sign up now](#)

Home

## Sign up

Already have an account? [Log in here](#)The website  
Name

signup/login

Home

Sign up

First Name:

Last Name:

Email Address:

Password:

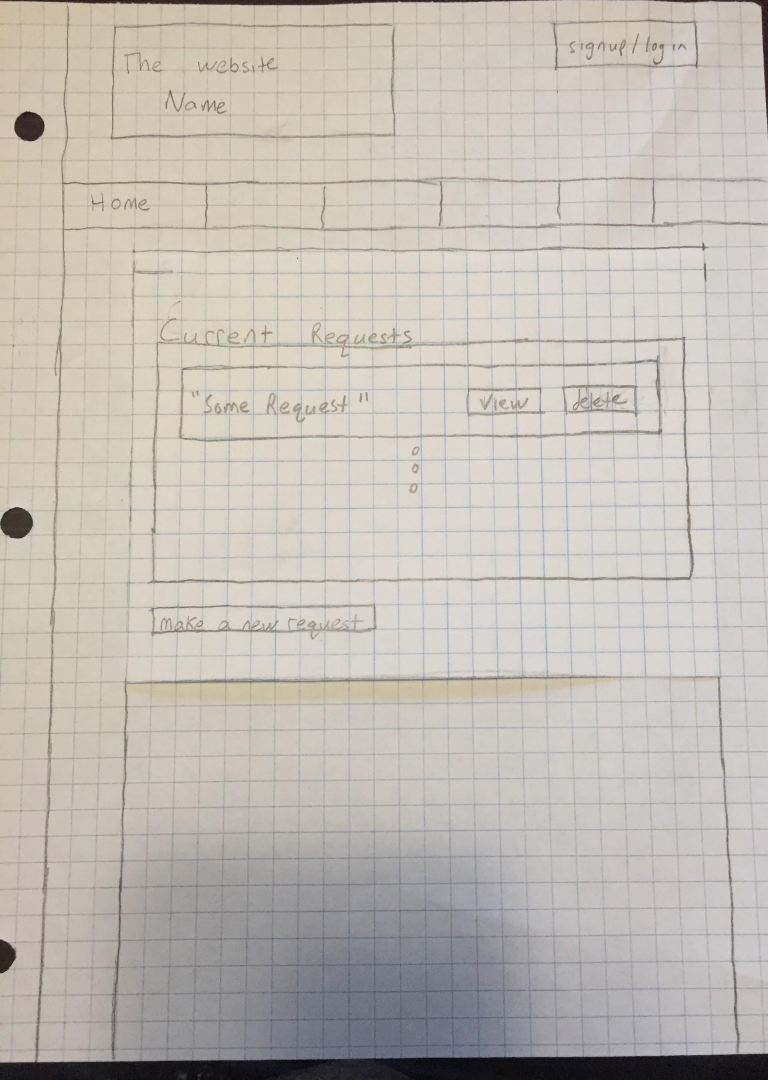
Verify Password:

Invite Code:

Sign up

Reset





## Open Requests

[New Request](#)

Approved

[View](#)

Approved

[View](#)

Approved

[View](#)

Approved

[View](#)

Denied

[View](#)

Denied

[View](#)

Approved

[View](#)

Pending

[View](#)

[Home](#)[My Requests](#)[Tickets](#)

## Request Software

[Cancel](#)[Request](#)

The website  
Name

signup/login

Home

Filters

1

Filter

2

List of programs

✓

Select



[Home](#)[My Requests](#)

## Ticket

Ticket state: Approved

Approve by: Joe Davis

Opened at: 2018-03-29T11:40:54.686Z

Last Updated at: 2018-03-29T13:20:07.279Z

Activate code: DASD-1ED1-VCUD-SI3N-DIAM

[Approve](#)[Deny](#)

The website  
Name

signup/login

Home

Ticket

First Name: " "

Last Name: " "

Email address: " "

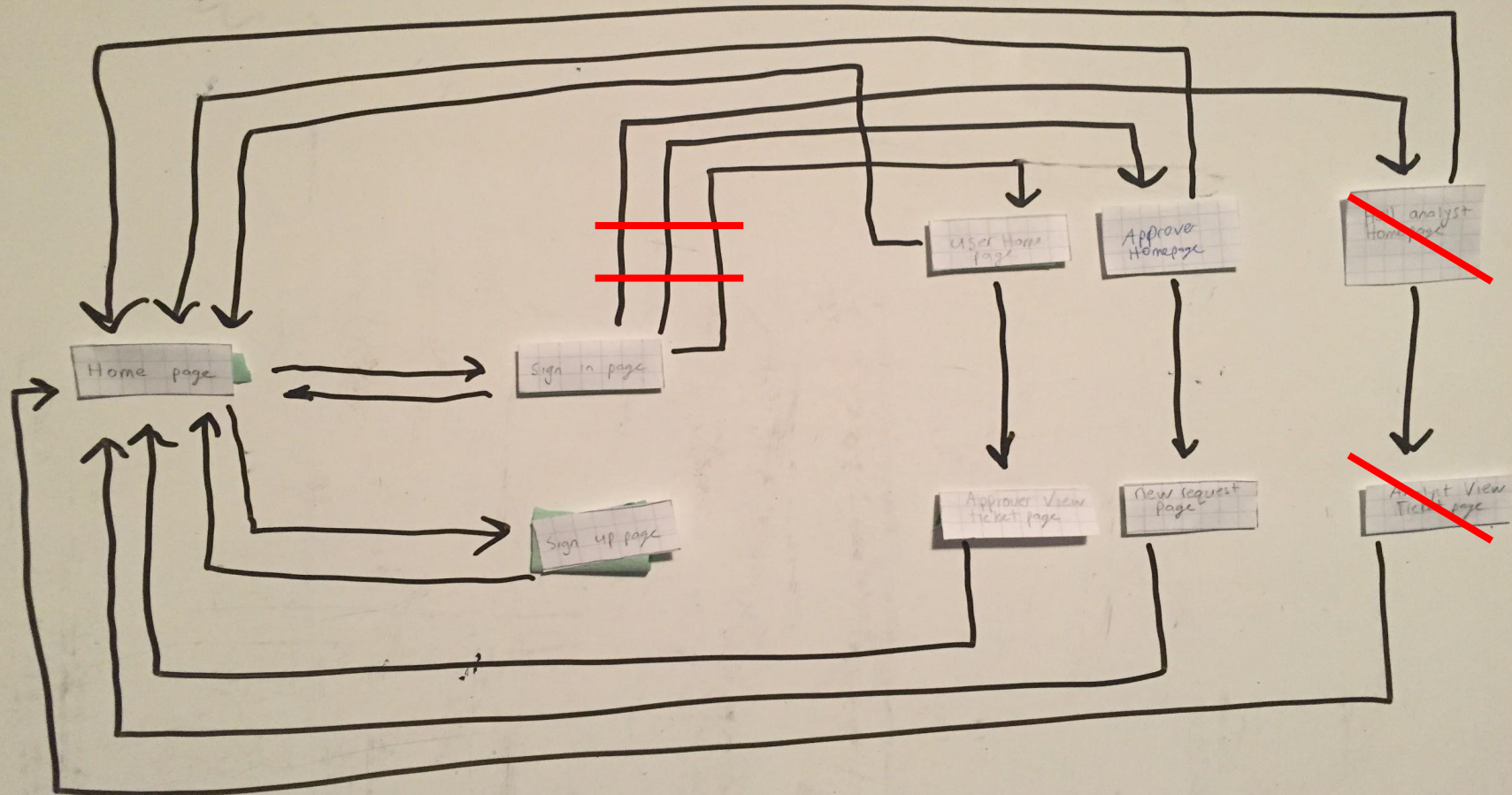
Company: " "

Active code: XXXX-XXXX-XXXX-XXXX

Ticket for approval

Approve

Decline





# Design Patterns

**State** - Vuex persists state such as the logged in user across the entire application

**Singleton** - Vuex \$store, there can only be one real truth

**Decorator** - Vue Router <router-view>, actual content and functionality determined at runtime (maybe a little abstract) and capable of being switched out at any time

**Composite** - Entire application is built as smaller components pieced together

**Mediator** - Services on the frontend contain the information that frontend components need to communicate with the server, though none of the frontend knows how the backend works

**Observer** - Again, Vuex mutates the \$store which is then broadcasted to the entire app, allowing whoever needs the state to deal with it as they wish



# Agile

- Only one sprint, didn't feel like we get to fully experience the full Agile process
- Would've liked to move stuff around a little to get two smaller sprints in rather than essentially one "big bang" sprint
  - Even though it was the MVP, it was also the entirety of what we needed to do which made it feel like a "big bang" approach
  - Missing out on the review section
- Overall, the plan, design, build aspect was great for organizing our work
  - The design phase felt less useful as our inexperience with the tech stack lead to many aspects being redesigned on the run; one more reason for two sprints

#	Acceptance test-driven development criteria	Failed	Passed
1	<i>Given the software requester has access to the internet, when they sign up for the website, then they will be able to view software requests s/he made before.</i>		✓
2	<i>Given the software requester is logged into the website, when they fill out and submit new request form, then they will see a ticket has been created.</i>		✓
3	<i>Given the software approver is logged into the website, when they look at the page display all tickets needing their action, then will be able to approve or deny the request.</i>		✓
4	<i>Given the software analyst is logged into the website, when they look at the page display all tickets needing their action, then will be able to approve or deny the request.</i>		✓
5	<i>Given the software analyst is logged in the application, when s/he looking at the request list, s/he is not able to preview the request without click on it.</i>	x (Doped)	
6	<i>Given the software analyst is logged in the application, when s/he looking at the top right of the list, then s/he will see the number of pending requests.</i>	x (2nd release)	
7	<i>Given the software analyst is processing the ticket, when s/he proceed the ticket and if the ticket was approved, then the system will notice user with a activated key.</i>	x (2nd release)	
8	<i>Given the user receive a notice of the ticket is completed, when s/he finalize the ticket, then s/he can confirm the ticket is complete or s/he can take additional actions on the ticket.</i>	x (2nd release)	





# **Demonstration**



# Group reflection

**How did we feel about this milestone? What did we like? What did we dislike?**

- It feels way more applicable than most other courses that we've had.

**What did you learn about yourself as you collaborated and worked through this milestone?**

- All-nighters become progressively less fun as your database has a full meltdown at 5am the morning it's due...

**How will you use what you have learned going forward?**

- The practical experience of building an application with a modern stack is invaluable for our futures. Perhaps a little skeptical of the real-life usage of the formal acceptance testing papers/methods.

**What “stuff & things” related to this milestone would you want help with?**

- Nothing, more time maybe.