Final Report

for

Rune

Version 1.0

UCLA CS 130

3 June 2016

Brandon Ly (304-136-729)
Justin Hou (204-155-681)
Roland Zeng (204-150-508)
Alex Wang (103-889-014)
Brendan Sio (804-150-223)
Kevin Tong (704-161-137)
Kevin Zuo (104-201-160)

1. Introduction
   1.1. Problem-Motivation
   1.2. Existing Solutions
   1.3. Rune Solution
2. Software Description
   2.1. Functional Requirements
      2.1.1. User Profiles
      2.1.2. Project Tracker
      2.1.3. Task Tracker
      2.1.4. Project Finder
   2.2. Nonfunctional Requirements
      2.2.1. Performance Requirements
      2.2.2. Safety and Security Requirements
      2.2.3. Software Quality Attributes
   2.3. Use Cases
   2.4. System Architecture (Diagram)
   2.5. Core Component Design (Diagram)
3. Development Requirements Verification
   3.1. Functional Requirements Verification
   3.1.1. Profiles
      3.1.2. Projects
      3.1.3. Tasks
      3.1.4. GitHub Integration
      3.1.5. Project Finder
   3.2. Non-Functional Requirements
      3.2.1. Performance
      3.2.2. Security
      3.2.3. Robustness
4. Development Challenges
   4.1. Modularity
   4.2. Code Reliability
5. Future Work
   5.1. Email Notifications
   5.2. Project Metrics
6. References

1. Introduction

1.1. Problem-Motivation

Rune, the Kanban Task Tracking Software, is envisioned to be a web-application software that resolves issues pertaining to software engineering development at UCLA. Throughout the course of his studies, a student majoring in Computer Science at UCLA will encounter many group Projects ranging from two to ten people, such as Projects for CS111 (pairs), CSM117 (3-5 people), and CS130 (4-10 people, depending on the professor). With this in mind, Rune aims to expedite software development in groups and to clarify communication between group members. The Kanban Task Tracking Software intends to answer questions that arise in group settings, such as who works on what, how long a Task has been in progress, and what Tasks are more important.

1.2. Existing Solutions

Currently, product management software such as JIRA, a proprietary issue tracking system for commercial organizations, and Trello, a dynamic virtual sticky board, exist. These tools, however, are either paid services that college students cannot afford, or too general and not specific enough to use for tracking software development Tasks. JIRA is utilized mainly by Agile development teams based that use the Scrum Agile framework. It provides users with the ability to keep track of Tasks with Scrum boards and Kanban boards, real-time Agile reporting, and portfolio planning.

1.3. Rune Solution

Rune is a Task-tracking management tool designed for students. It is heavily inspired by JIRA, with particular features finely-tuned to cater towards university students working on software Projects. The primary feature of Rune is the use of the of the Kanban methodology and the Kanban Board for Task management in Projects (compared to JIRA, which focuses on the Agile and Scrum methodology). Rune is a web application; students use the client through a modern web browser and the client interacts with a server stack to maintain data on Rune.

2. Software Description

2.1. Functional Requirements

2.1.1. User Profiles

Rune shall have a user profile system, with each user profile containing details about users of the application. User profiles essentially contribute to authorization mechanisms for Projects and Tasks in the application. User profiles shall have the following details that users can choose to fill out, are required to fill out, or will already be filled out:

- Username (email) (required)
- First and Last Name (required)
- Password (required)
- Description Field
- Skills Field
- GitHub Profile

2.1.2. Project Tracker

The Project tracker follows all the Projects that are created with Rune. It provides the users with details about their Project, including associated users, etc. Projects shall have the following detail components:

- Project Name (required)
- Project Key (required)
- Project ID (required)
- Admin User (required)
- Associated User(s)
- Associated Task(s)
- Description (required)
- History
- GitHub Repository

Project details shall also be fully editable by users who are a part of the Project.

2.1.3. Task Tracker

The Task tracker follows all Tasks that are created with Rune. It provides users with details about a Task, such as it's description, the category (i.e. Task, feature), the development stage, and comment log. Tasks shall have the following detail components:

- Task Name (required)
- Task ID (required)
- Associated Project (required)

- Task Description (required)
- Assignee (required)
- Status (To-do, In Progress, Code Review, Testing, Completed) (required)
- Priority (required)
- Comment Log

Task details shall be fully editable by users who have access to its associated Project.

2.1.4. Project Finder

Rune shall have a Project Finder, a page listing all Projects on the application so that other students can find Projects based on similar interests. Users can apply to Projects, and the Admin User, after inspecting the applicant's profile, can add them to the Project. Projects on the Project Finder listings shall have, including the specified Project details, the following details:

- Project Public/Private Setting (required)
- Applicants Pending (required)
- Admin User (required)
- Project Description (required)
- Project Skills (required)
- Associated User Count (required)

2.2. Nonfunctional Requirements

Nonfunctional requirements of Rune include performance, security, and robustness. Pages in the application must load in a reasonable amount of time. Data created by users along with their information should be secure.

2.2.1. Performance Requirements

Rune has two primary performance requirements, including client-side performance requirements and server-side performance requirements.

All server-side interactions with the client-side features on the application shall perform as well as most commercial web services. This means that, on optimal internet connection speeds, Rune shall run server-side operations (i.e. database queries) with a runtime of ~0.1 s average, <5.0s maximum.

All client-side interactions on the application shall perform as well as most commercial web applications. This means that, on optimal internet connection speeds, Rune shall take ~0.5 s on average, <8.0s at worst to load new pages and dynamic elements on the page. This cascades if certain client-side interactions involve server-side operations; time to perform client-side operations includes the time that each server-side operation takes. Also, all dynamic elements of any page shall run smoothly on at least 60 Hz (the user shall not experience any "lagginess" when using the application).

2.2.2. Safety and Security Requirements

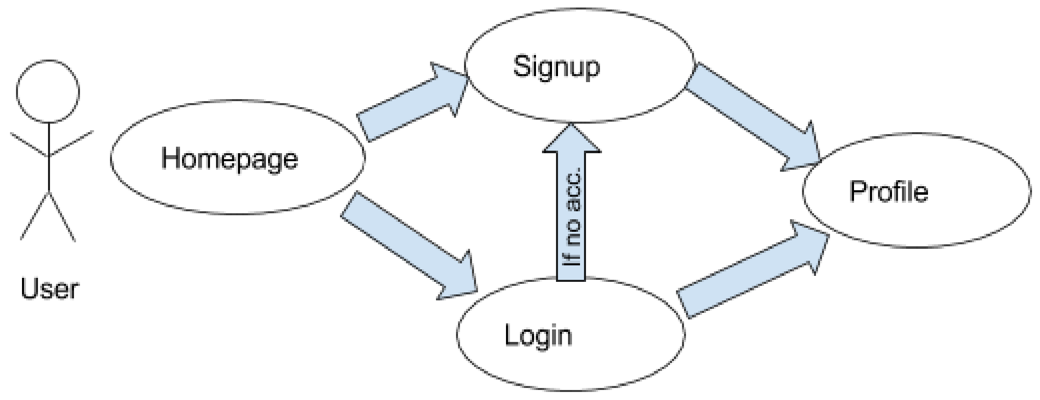
Rune shall be as safe to use as any other web application (i.e. safety is therefore up to the users themselves). The web application shall use standard security technology and protocols (i.e. HTTPS, SSH, OpenSSL, etc) to ensure that the application and it's data will most likely not be compromised.

2.2.3. Software Quality Attributes

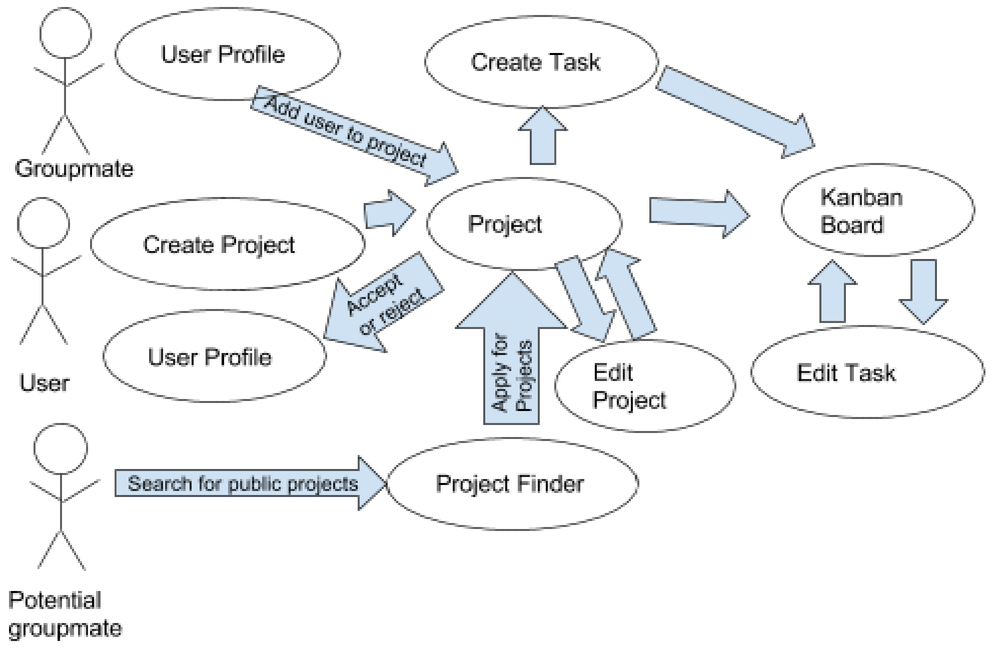
Rune its user experience design, and its software architecture shall follow the highest standards of design and software engineering. User interfaces shall make the user experience as easy to use and nice to look at as possible; it shall be designed with ease of usability in mind. Written code shall be cleanly formatted and well-commented and shall follow standard object-oriented principles. The architecture of the codebase shall also be scalable; increasing the scale and adapting new features shall be straightforward to implement.

2.3. Use Cases

The following diagram visualizes a user login use case:

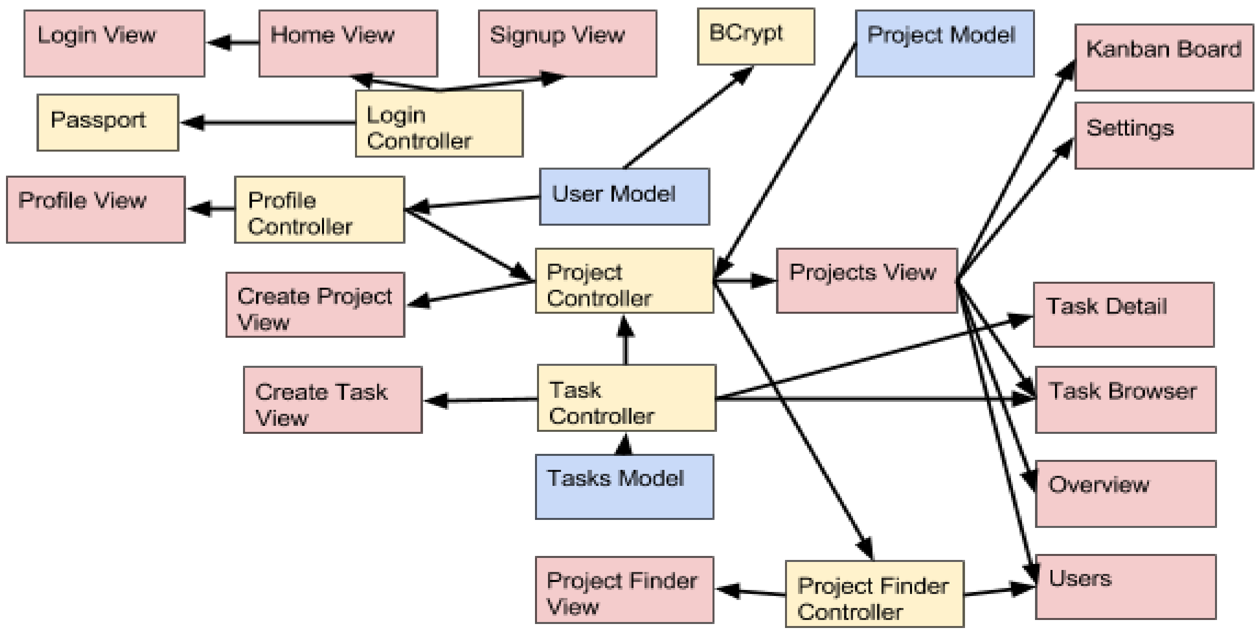


The following diagram visualizes the use cases of Project creation and Task creation:



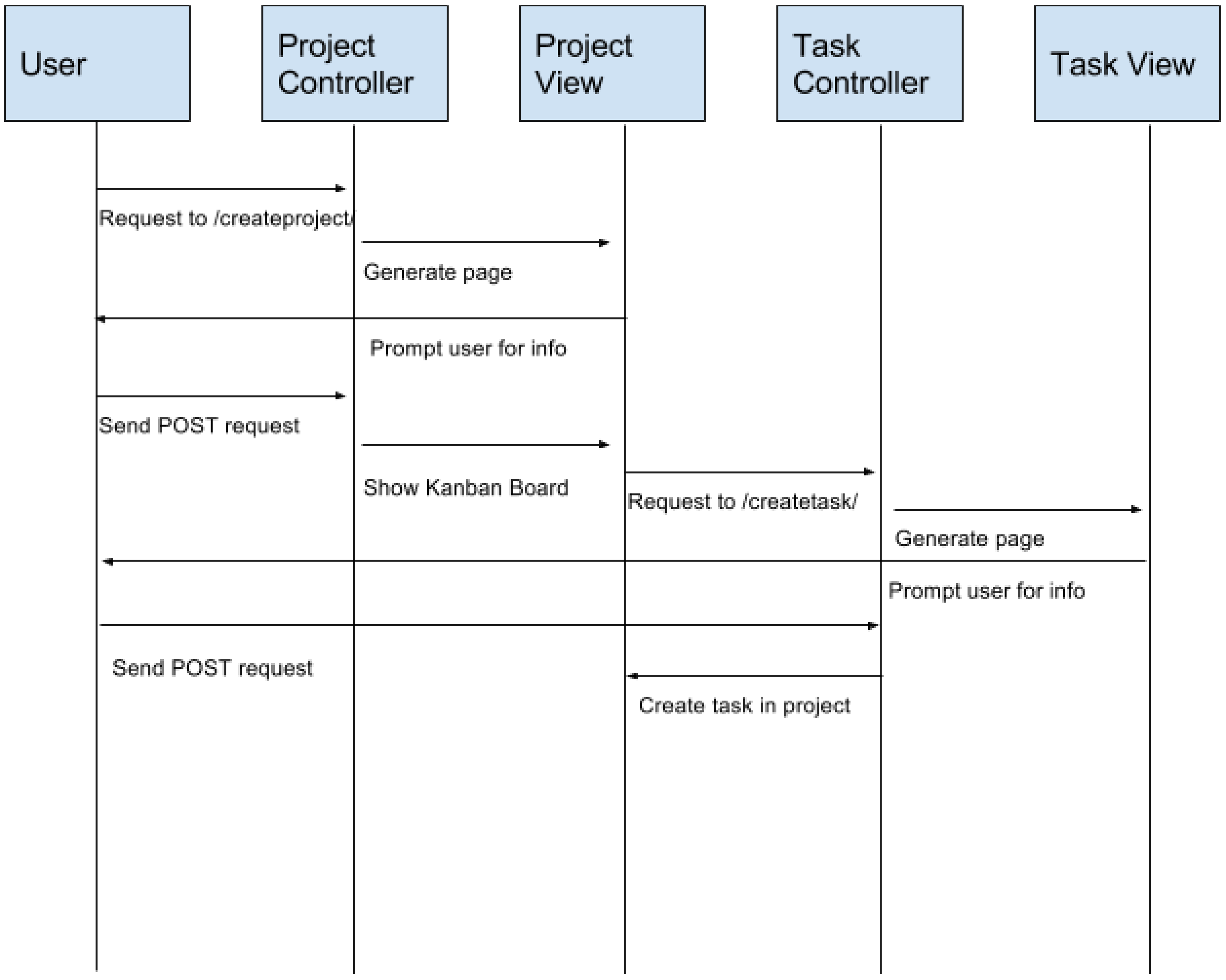
2.4. System Architecture (Diagram)

Rune follows the Model-View-Controller architectural pattern. In the following diagram, red boxes represent view, blue boxes represent models, and yellow boxes represent controllers.



2.5. Core Component Design (Diagram)

The following diagram shows the various interactions between the core components of Rune:

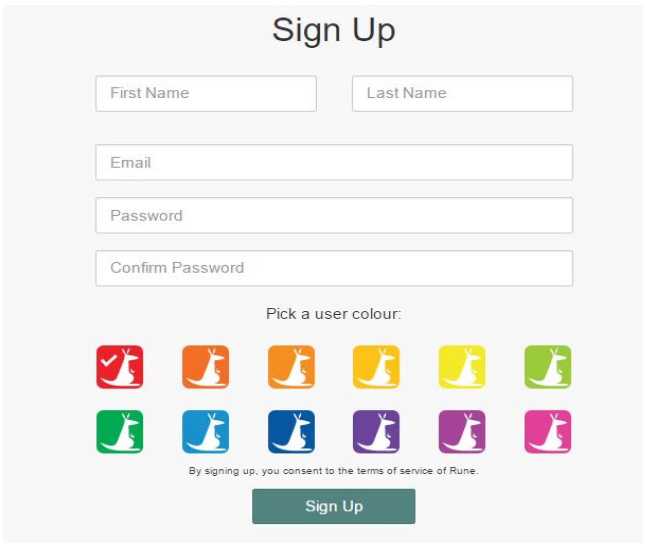


3. Development Requirements Verification

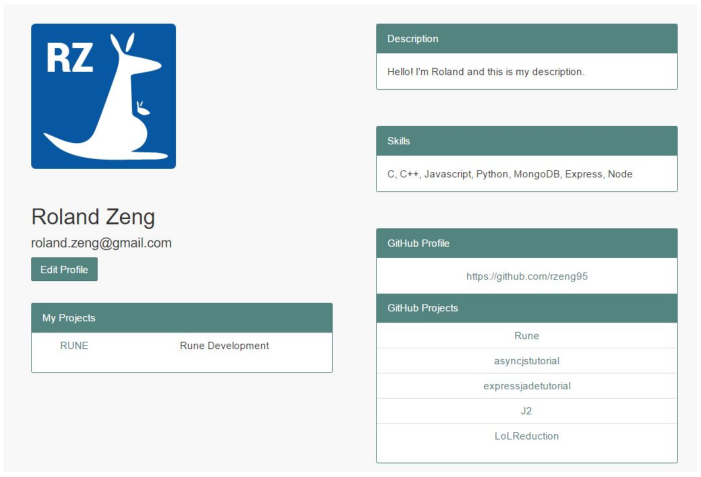
3.1. Functional Requirements Verification

3.1.1. Profiles

User profiles are generated upon successful signups. Signups are created by entering in a unique email, a first and last name, a password, and a profile picture color. Below is the completed, functional signup form.



Successful signups redirect users to their profile page. The profile page contains publicly available information about a user, allowing for editable fields for a user's description and skillset. Rune Projects that the user is a member of are displayed on the profile page. Furthermore, users are given the ability to link their Rune accounts with their GitHub accounts. Users who perform this synchronization are able to display a list of their most recent GitHub Projects. Altogether, a user's profile page looks like the following:



Roland Zeng
roland.zeng@gmail.com

My Projects

RUNE	Rune Development

Description

Hello! I'm Roland and this is my description.

Skills

C, C++, Javascript, Python, MongoDB, Express, Node

GitHub Profile

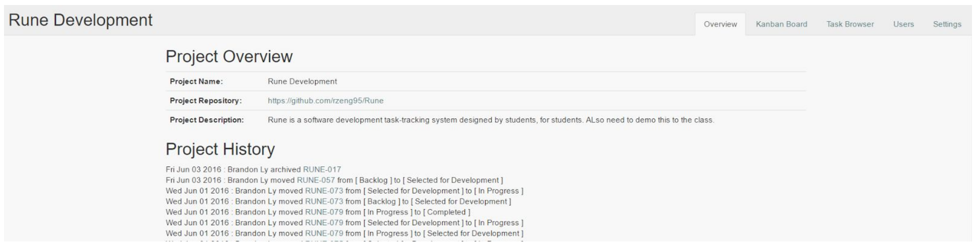
https://github.com/rzeng95

GitHub Projects

Rune
asyncjstutorial
expressjadetutorial
J2
LoLReduction

3.1.2. Projects

Each Rune Project is given a unique page. Each Project page contains five tabs. The first tab, the Overview, displays basic information about the Project as well as the Project history, a record of Task creations, modifications, and deletions.



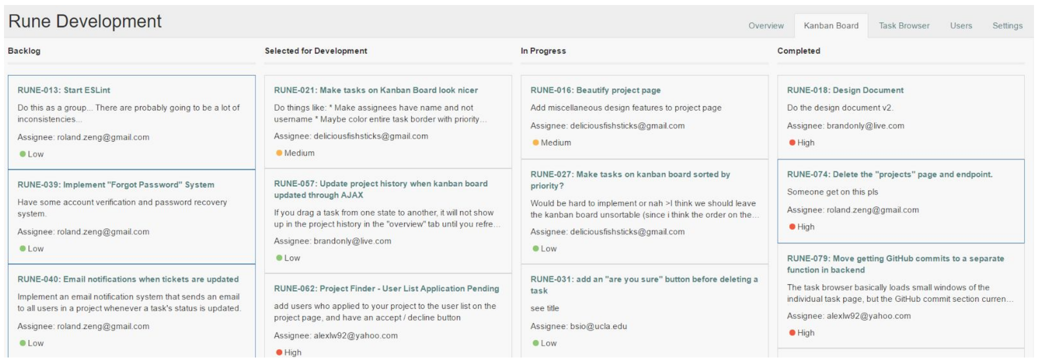
Project Overview

Project Name:	Rune Development
Project Repository:	https://github.com/rzeng95/Rune
Project Description:	Rune is a software development task-tracking system designed by students, for students. ALso need to demo this to the class.

Project History

Fri Jun 03 2016 : Brandon Ly archived RUNE-017
Fri Jun 03 2016 : Brandon Ly moved RUNE-057 from [Backlog] to [Selected for Development]
Wed Jun 01 2016 : Brandon Ly moved RUNE-073 from [Selected for Development] to [In Progress]
Wed Jun 01 2016 : Brandon Ly moved RUNE-073 from [Backlog] to [Selected for Development]
Wed Jun 01 2016 : Brandon Ly moved RUNE-079 from [In Progress] to [Completed]
Wed Jun 01 2016 : Brandon Ly moved RUNE-079 from [Selected for Development] to [In Progress]
Wed Jun 01 2016 : Brandon Ly moved RUNE-079 from [In Progress] to [Selected for Development]

The second tab, the Kanban board, is the "meat" of each Project page. It displays a Project's active Tasks on a dynamic virtual sticky Kanban board, where the Tasks can be modified by dragging and dropping them on the board between four statuses: Backlog, Selected for Development, In Progress, and Completed. Clicking on each Task should link it to the corresponding Task page.



Backlog

RUNE-013: Start ESLint
Do this as a group... There are probably going to be a lot of inconsistencies...
Assignee: roland.zeng@gmail.com
● Low

RUNE-039: Implement "Forgot Password" System
Have some account verification and password recovery system.
Assignee: roland.zeng@gmail.com
● Low

RUNE-040: Email notifications when tickets are updated
Implement an email notification system that sends an email to all users in a project whenever a task's status is updated.
Assignee: roland.zeng@gmail.com
● Low

Selected for Development

RUNE-021: Make tasks on Kanban Board look nicer
Do things like: * Make assignees have name and not username * Maybe color entire task border with priority...
Assignee: deliciousfishsticks@gmail.com
● Medium

RUNE-057: Update project history when kanban board updated through AJAX
If you drag a task from one state to another, it will not show up in the project history in the "overview" tab until you refre...
Assignee: brandonly@live.com
● Low

RUNE-062: Project Finder - User List Application Pending
add users who applied to your project to the user list on the project page, and have an accept / decline button
Assignee: alexlw92@yahoo.com
● High

In Progress

RUNE-016: Beautify project page
Add miscellaneous design features to project page
Assignee: deliciousfishsticks@gmail.com
● Medium

RUNE-027: Make tasks on kanban board sorted by priority?
Would be hard to implement or nah >I think we should leave the kanban board unsortable (since i think the order on the...
Assignee: deliciousfishsticks@gmail.com
● Low

RUNE-031: add an "are you sure" button before deleting a task
see title
Assignee: bsio@ucla.edu
● Low

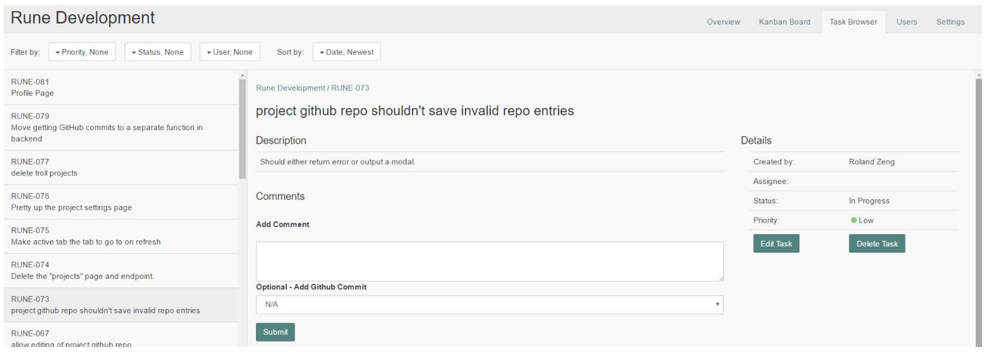
Completed

RUNE-018: Design Document
Do the design document v2.
Assignee: brandonly@live.com
● High

RUNE-074: Delete the "projects" page and endpoint.
Someone get on this pls
Assignee: roland.zeng@gmail.com
● High

RUNE-079: Move getting GitHub commits to a separate function in backend
The task browser basically loads small windows of the individual task page, but the GitHub commit section curren...
Assignee: alexlw92@yahoo.com
● High

The third tab, the Task Browser, lists every single Task created for the Project. This includes archived Tasks, which are Tasks that have been removed from the Kanban board but not deleted. The Task Browser is sortable and filterable by priority, members, status, and date. These features have been implemented and are displayed below.



Filter by: Priority, None | Status, None | User, None   Sort by: Date, Newest

RUNE-081
Profile Page

RUNE-079
Move getting GitHub commits to a separate function in backend

RUNE-077
delete troll projects

RUNE-076
Pretty up the project settings page

RUNE-075
Make active tab the tab to go to on refresh

RUNE-074
Delete the "projects" page and endpoint.

RUNE-073
project github repo shouldn't save invalid repo entries

RUNE-067
allow editing of project github repo

Rune Development / RUNE-073

project github repo shouldn't save invalid repo entries

Description

Should either return error or output a modal.

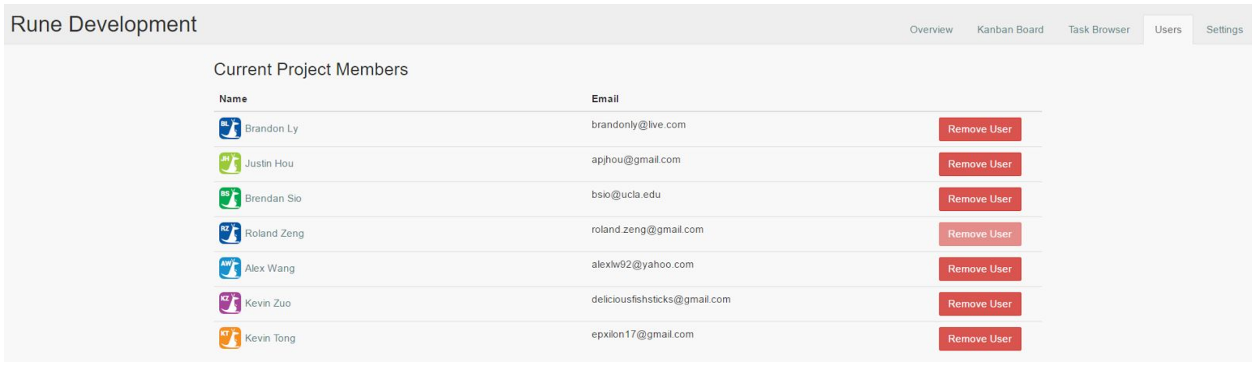
Comments

Optional - Add Github Commit: N/A

Details

Created by:	Roland Zeng
Assignee:	
Status:	In Progress
Priority:	● Low

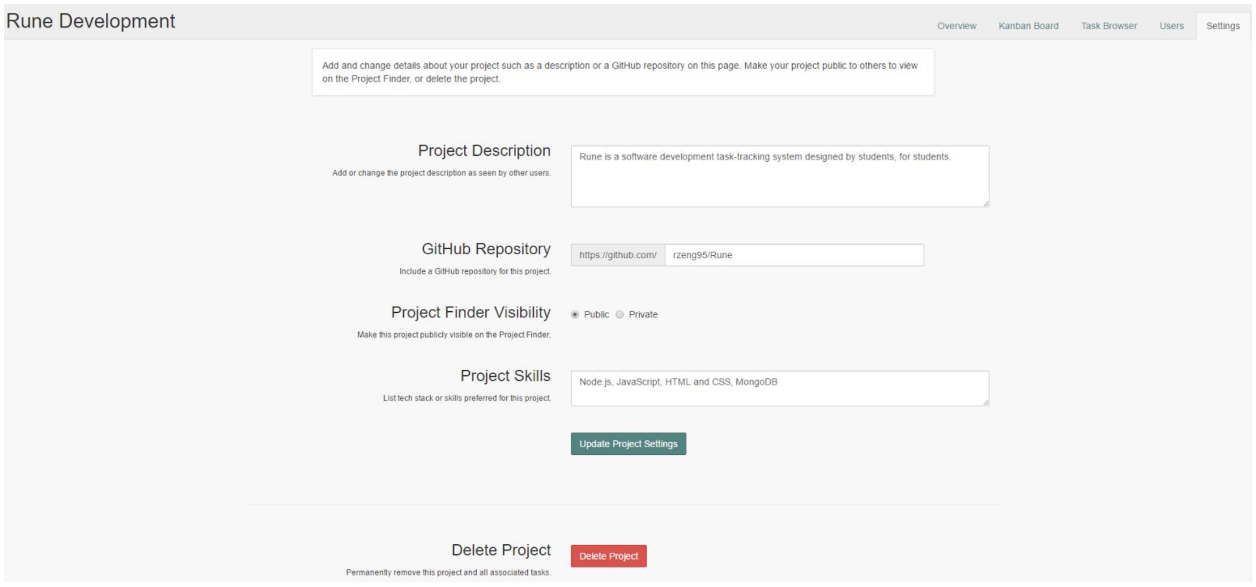
The fourth tab is the users list, which displays the list of a Project's members. Members should be removable by the Project creator. Furthermore, Users that have applied to join the Project should be displayed, and can be accepted by Project creators (not pictured).



Current Project Members

Name	Email	
Brandon Ly	brandonly@live.com	Remove User
Justin Hou	apjhou@gmail.com	Remove User
Brendan Sio	bsio@ucla.edu	Remove User
Roland Zeng	roland.zeng@gmail.com	Remove User
Alex Wang	alexlw92@yahoo.com	Remove User
Kevin Zuo	deliciousfishsticks@gmail.com	Remove User
Kevin Tong	epxilon17@gmail.com	Remove User

The fifth tab is the Project settings tab. This allows editing of a Project's description, skills, and GitHub repository. The Project can be listed on the Project Finder by setting Visibility to public. Finally, Projects can be deleted by the Project owner (following a confirmation message).



Add and change details about your project such as a description or a GitHub repository on this page. Make your project public to others to view on the Project Finder, or delete the project.

Project Description
Add or change the project description as seen by other users.
Rune is a software development task-tracking system designed by students, for students.

GitHub Repository
Include a GitHub repository for this project.
https://github.com/ rzeng95/Rune

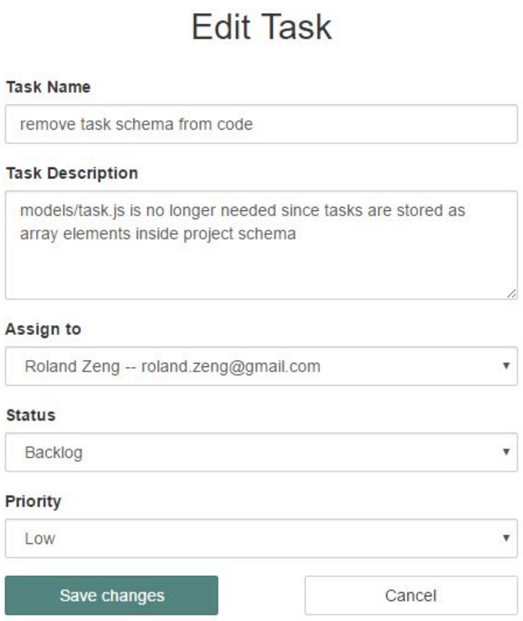
Project Finder Visibility
Make this project publicly visible on the Project Finder.
[✓ Public] Private

Project Skills
List tech stack or skills preferred for this project.
Node.js, JavaScript, HTML and CSS, MongoDB

Delete Project
Permanently remove this project and all associated tasks.

3.1.3. Tasks

Tasks are created via the Create Task button within a Project page. Tasks are given a name, a description, an assignee, a status, and a priority. The status of a Task determines its position on the Kanban board. When Tasks are marked as complete, the option to archive a Task is given. Archived Tasks are removed from the Kanban board but preserved in the Task browser. Users can add comments to each Task. Tasks can also be deleted.



Edit Task

Task Name

remove task schema from code

Task Description

models/task.js is no longer needed since tasks are stored as array elements inside project schema

Assign to

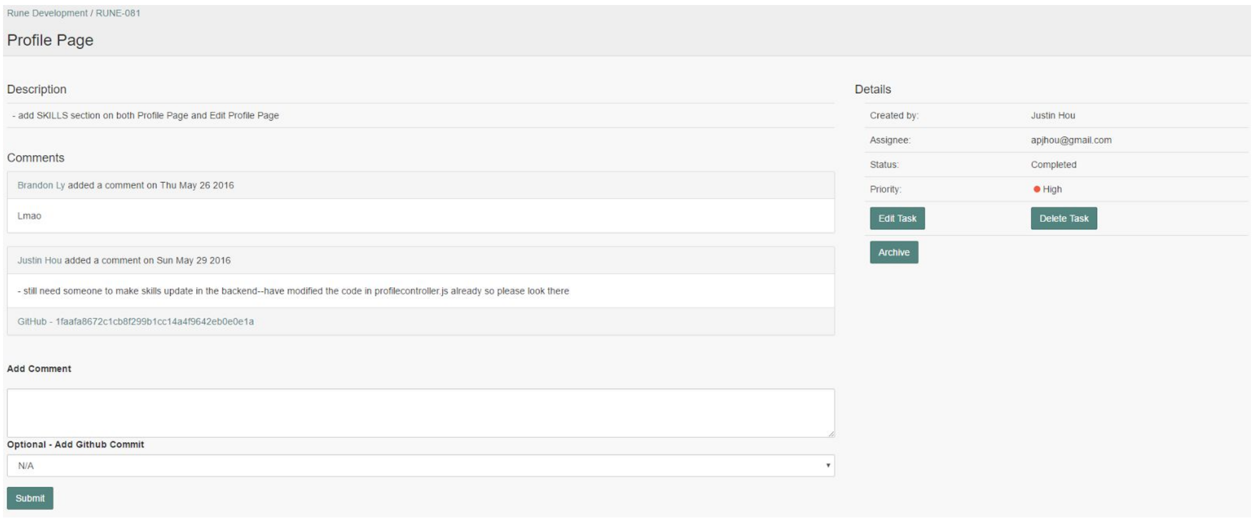
Roland Zeng -- roland.zeng@gmail.com

Status

Backlog

Priority

Low



Rune Development / RUNE-081

Profile Page

Description

- add SKILLS section on both Profile Page and Edit Profile Page

Comments

Brandon Ly added a comment on Thu May 26 2016

Lmao

Justin Hou added a comment on Sun May 29 2016

- still need someone to make skills update in the backend--have modified the code in profilecontroller.js already so please look there

GitHub - 1faafa8672c1cb8f299b1cc14a4f9642eb0e0e1a

Add Comment

Optional - Add Github Commit

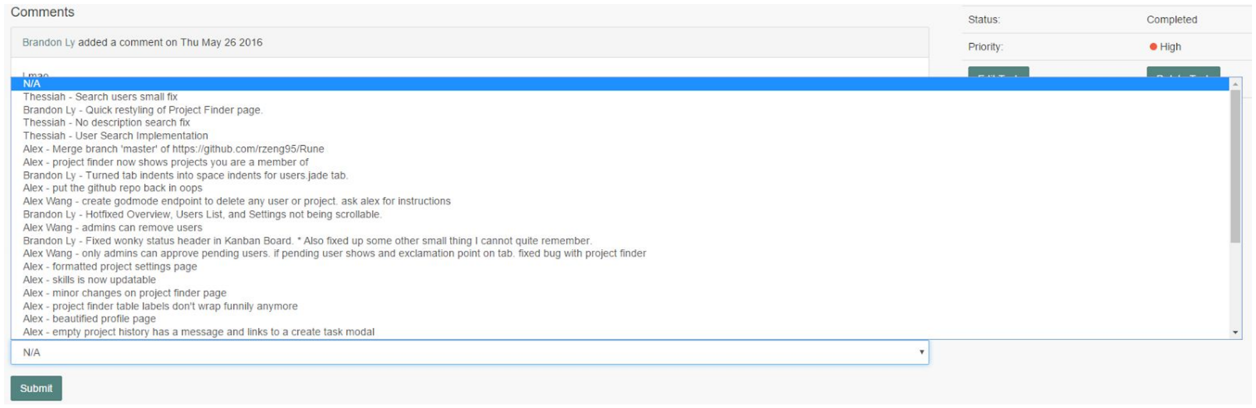
N/A

Details

Created by:	Justin Hou
Assignee:	apjhou@gmail.com
Status:	Completed
Priority:	● High

3.1.4. GitHub Integration

Projects can be linked to GitHub Project repositories. When a user adds a comment to a Task, they have the option to associate the Task with one of the Project's GitHub commits. This feature is hidden for Rune Projects whose owners have not entered a GitHub Project repository name.



Comments

Brandon Ly added a comment on Thu May 26 2016

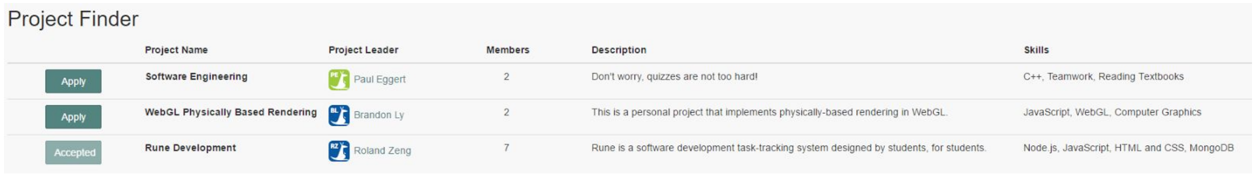
Status: Completed
Priority: ● High

```
N/A
Thessiah - Search users small fix
Brandon Ly - Quick restyling of Project Finder page.
Thessiah - No description search fix
Thessiah - User Search Implementation
Alex - Merge branch 'master' of https://github.com/rzeng95/Rune
Alex - project finder now shows projects you are a member of
Brandon Ly - Turned tab indents into space indents for users.jade tab.
Alex - put the github repo back in oops
Alex Wang - create godmode endpoint to delete any user or project. ask alex for instructions
Brandon Ly - Hotfixed Overview, Users List, and Settings not being scrollable.
Alex Wang - admins can remove users
Brandon Ly - Fixed wonky status header in Kanban Board. * Also fixed up some other small thing I cannot quite remember.
Alex Wang - only admins can approve pending users. if pending user shows and exclamation point on tab. fixed bug with project finder
Alex - formatted project settings page
Alex - skills is now updatable
Alex - minor changes on project finder page
Alex - project finder table labels don't wrap funnily anymore
Alex - beautified profile page
Alex - empty project history has a message and links to a create task modal
```

N/A

3.1.5. Project Finder

Project Finder allows users to search for Projects to join. Only publicly-listed Projects are listed, and users can apply to any Project they are not a member of. Each Project displays its creator, member count, description, and skills.



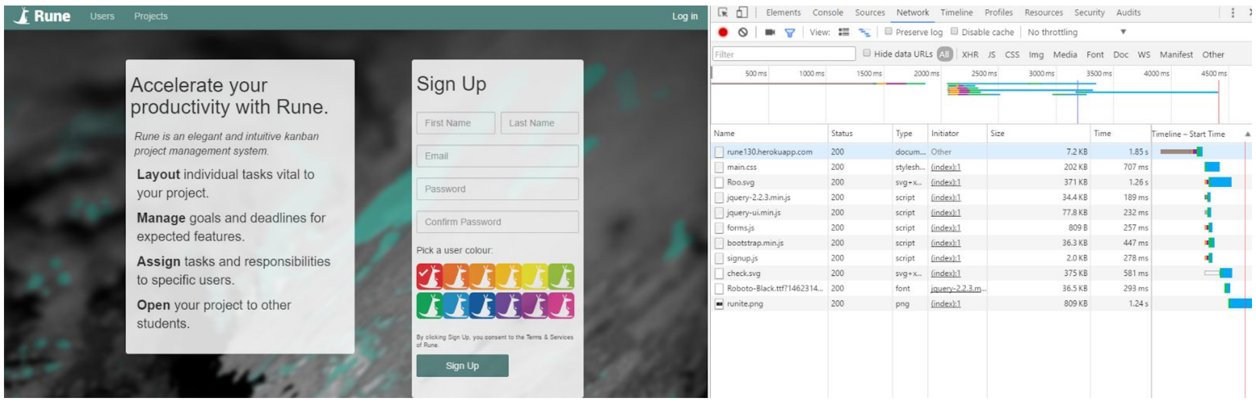
Project Finder

	Project Name	Project Leader	Members	Description	Skills
Apply	Software Engineering	Paul Eggert	2	Don't worry, quizzes are not too hard!	C++, Teamwork, Reading Textbooks
Apply	WebGL Physically Based Rendering	Brandon Ly	2	This is a personal project that implements physically-based rendering in WebGL.	JavaScript, WebGL, Computer Graphics
Accepted	Rune Development	Roland Zeng	7	Rune is a software development task-tracking system designed by students, for students.	Node.js, JavaScript, HTML and CSS, MongoDB

3.2. Non-Functional Requirements

3.2.1. Performance

Performance varied depending on page content. The homepage took about 4 seconds to initially load, with subsequent loads displaying in less than 1 second, due to browser-side caching. The user profile page loaded within 1.5 seconds, due to having significantly less content to display. The Project page for Rune Development initially loaded in 2.5 seconds, with subsequent loads not affecting the load time. This was most likely due to the sheer amount of text that had to be passed to the view from the back-end. In the future, faster load times can be achieved by implementing server-side or client-side caching, as many Tasks are unchanged when a Project is viewed, but need only be rendered visible on the Kanban board.



Accelerate your productivity with Rune.

Rune is an elegant and intuitive kanban project management system.

Layout individual tasks vital to your project.

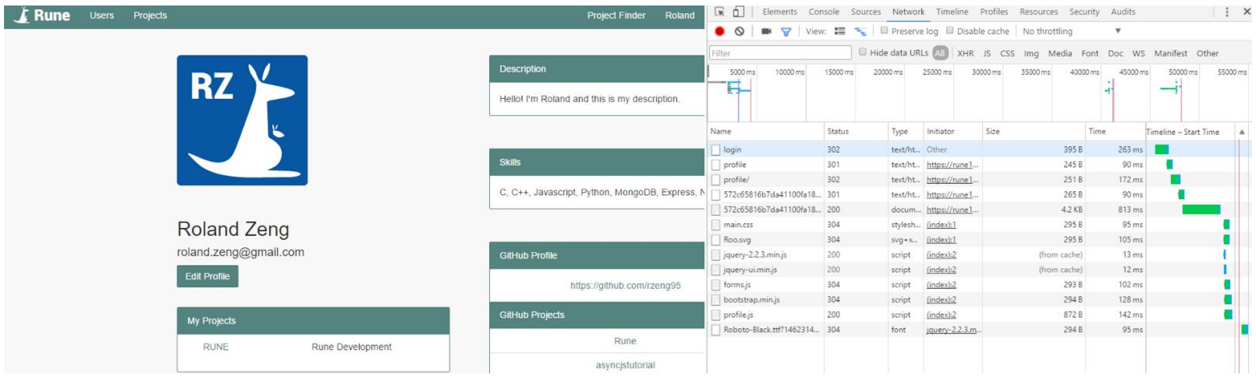
Manage goals and deadlines for expected features.

Assign tasks and responsibilities to specific users.

Open your project to other students.

By clicking Sign Up, you consent to the Terms & Services of Rune.

Name	Status	Type	Initiator	Size	Time
rune130.herokuapp.com	200	docum...	Other	7.2 KB	1.85 s
main.css	200	stylesh...	(index):1	202 KB	707 ms
Roo.svg	200	svg+x...	(index):1	371 KB	1.26 s
jquery-2.2.3.min.js	200	script	(index):1	34.4 KB	189 ms
jquery-ui.min.js	200	script	(index):1	77.8 KB	232 ms
forms.js	200	script	(index):1	809 B	257 ms
bootstrap.min.js	200	script	(index):1	36.3 KB	447 ms
signup.js	200	script	(index):1	2.0 KB	278 ms
check.svg	200	svg+x...	(index):1	375 KB	581 ms
Roboto-Black.ttf?1462314...	200	font	jquery-2.2.3.m...	36.5 KB	293 ms
runite.png	200	png	(index):1	809 KB	1.24 s



Roland Zeng
roland.zeng@gmail.com

Hello! I'm Roland and this is my description.

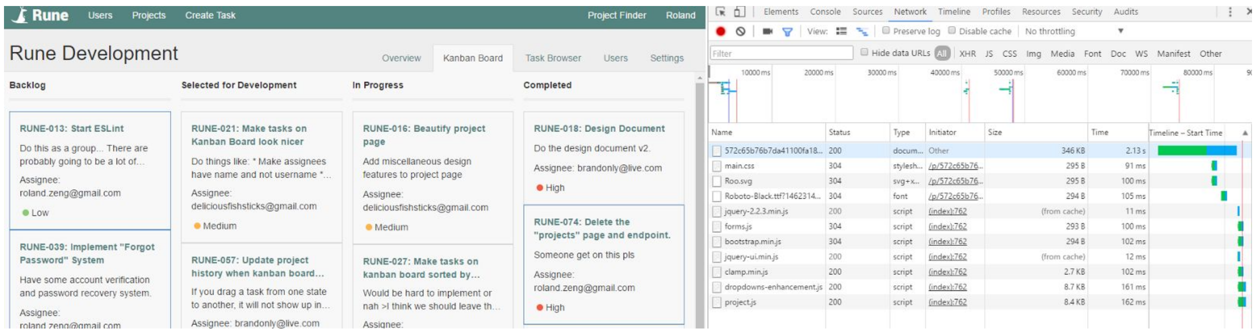
C, C++, Javascript, Python, MongoDB, Express, N

https://github.com/rzeng95

Rune
asyncjstutorial

RUNE Rune Development

Name	Status	Type	Initiator	Size	Time
login	302	text/ht...	Other	395 B	263 ms
profile	301	text/ht...	https://rune1...	245 B	90 ms
profile/	302	text/ht...	https://rune1...	251 B	172 ms
572c65816b7da41100fa18...	301	text/ht...	https://rune1...	265 B	90 ms
572c65816b7da41100fa18...	200	docum...	https://rune1...	4.2 KB	813 ms
main.css	304	stylesh...	(index):1	295 B	95 ms
Roo.svg	304	svg+x...	(index):1	295 B	105 ms
jquery-2.2.3.min.js	200	script	(index):2	(from cache)	13 ms
jquery-ui.min.js	200	script	(index):2	(from cache)	12 ms
forms.js	304	script	(index):2	293 B	102 ms
bootstrap.min.js	304	script	(index):2	294 B	128 ms
profile.js	200	script	(index):2	872 B	142 ms
Roboto-Black.ttf?1462314...	304	font	jquery-2.2.3.m...	294 B	95 ms



Rune Development

Backlog

RUNE-013: Start ESLint
Do this as a group... There are probably going to be a lot of...
Assignee: roland.zeng@gmail.com
● Low

RUNE-039: Implement "Forgot Password" System
Have some account verification and password recovery system.
Assignee: roland.zeng@gmail.com

Selected for Development

RUNE-021: Make tasks on Kanban Board look nicer
Do things like: * Make assignees have name and not username *...
Assignee: deliciousfishsticks@gmail.com
● Medium

RUNE-057: Update project history when kanban board...
If you drag a task from one state to another, it will not show up in...
Assignee: brandonly@live.com

In Progress

RUNE-016: Beautify project page
Add miscellaneous design features to project page
Assignee: deliciousfishsticks@gmail.com
● Medium

RUNE-027: Make tasks on kanban board sorted by...
Would be hard to implement or nah >I think we should leave th...
Assignee:

Completed

RUNE-018: Design Document
Do the design document v2.
Assignee: brandonly@live.com
● High

RUNE-074: Delete the "projects" page and endpoint.
Someone get on this pls
Assignee: roland.zeng@gmail.com
● High

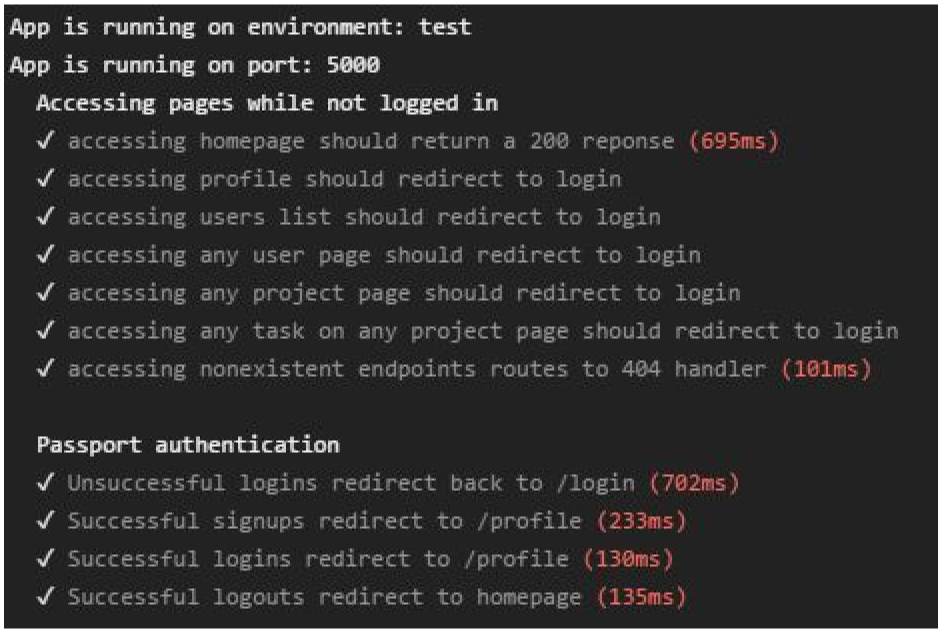
Name	Status	Type	Initiator	Size	Time
572c65b76b7da41100fa18...	200	docum...	Other	346 KB	2.13 s
main.css	304	stylesh...	/p/572c65b76...	295 B	91 ms
Roo.svg	304	svg+x...	/p/572c65b76...	295 B	100 ms
Roboto-Black.ttf?1462314...	304	font	/p/572c65b76...	294 B	105 ms
jquery-2.2.3.min.js	200	script	(index):762	(from cache)	11 ms
forms.js	304	script	(index):762	293 B	100 ms
bootstrap.min.js	304	script	(index):762	294 B	102 ms
jquery-ui.min.js	200	script	(index):762	(from cache)	12 ms
clamp.min.js	200	script	(index):762	2.7 KB	102 ms
dropdowns-enhancement.js	200	script	(index):762	8.7 KB	161 ms
project.js	200	script	(index):762	8.4 KB	162 ms

3.2.2. Security

Rune is deployed via Heroku, a cloud application platform. According to their security page, the Heroku platform is secured on many levels. Their security validation systems are their bug reporting bounty system and third party penetration teams. Their data centers are secured with climate control and fire detection and suppression. Data is separated by users and each account can only see data from their account. Heroku also follows Best Practices such as encrypting data in transit and at rest, use of third party systems, secure development practices, and logging. All of the security practices that Heroku uses ensures that Rune has secure and safe hosting.

3.2.3. Robustness

Code robustness was enforced by linter programs such as ESLint, that ensured that all members followed the same coding standards. For example, ESLint ensured that all lines of code ended with semicolons, and all strings were wrapped in single quotes. To ensure code reliability, automated tests were written using the Mocha testing framework. These tests were run every time new changes were checked into GitHub, to confirm that the new build did not break basic site functionality. This system of automated tests was handled by Travis-CI, a continuous integration platform.



```
App is running on environment: test
App is running on port: 5000
  Accessing pages while not logged in
  ✓ accessing homepage should return a 200 reponse (695ms)
  ✓ accessing profile should redirect to login
  ✓ accessing users list should redirect to login
  ✓ accessing any user page should redirect to login
  ✓ accessing any project page should redirect to login
  ✓ accessing any task on any project page should redirect to login
  ✓ accessing nonexistent endpoints routes to 404 handler (101ms)

  Passport authentication
  ✓ Unsuccessful logins redirect back to /login (702ms)
  ✓ Successful signups redirect to /profile (233ms)
  ✓ Successful logins redirect to /profile (130ms)
  ✓ Successful logouts redirect to homepage (135ms)
```

4. Development Challenges

Building on such a large code base presented many problems associated with coding with large software teams. Issues such as code understandability, merge conflicts, and code reliability plagued development, and are addressed below.

4.1. Modularity

Early on, it became necessary to modularize the rapidly-growing code base in order to improve understandability and reduce merge conflicts. This was done by first separating code into models, views, and controllers (MVC), and then separating controller endpoints by functionality. For example, all of the endpoints that dealt with Project creation and management were handled in a file called "ProjectController.js". This not only enabled logical separation of controller code, it also allowed for smaller code files and ease of understandability.

4.2. Code Reliability

As more changes were checked into the main build, the possibility of errors or bugs greatly increased. For example, modifying database schemas could have led to other controllers that accessed those database models to break. Previously, the primary method of assuring code quality was to run a local version of Rune prior to each deployment and make sure important pages loaded and functioned correctly. However, this method of manual testing was rather inefficient and prone to human error. This problem was solved by implementing a continuous integration pipeline using Travis-CI. Whenever new changes were checked to the build, they would be run against a series of automated tests. If all the tests successfully executed, then the build would be pushed to the live website. Otherwise, Travis CI would send an email to the committer to alert them that their change had broke the build. After implementing automated tests, it became much easier to establish code reliability.

5. Future Work

Rune was meant to offer a comprehensive Project management service for students. This meant first and foremost emulating enterprise-level tools such as Jira, and then focusing on creating additional features. Because of this development direction, many extra features could not be implemented in lieu of completing core features. These extra features are described below.

5.1. Email Notifications

Email notifications are crucial to keeping Project members updated on changes to the Project. Implementing email features using frameworks such as NodeMailer or EmailJS. Emails also could be used to implement email recovery, which any software must utilize. Because of possible future plans to integrate with UCLA Shibboleth, email recovery could possibly be implemented later.

5.2. Project Metrics

Project metrics includes information such as Tasks completed by each member, and Project velocity. Metrics were not implemented due to lack of experience working with data visualization frameworks such as D3JS.

6. References

"Heroku Security | Heroku." Heroku Security | Heroku. N.p., n.d. Web. 04 June 2016.

"JIRA Software - Issue & Project Tracking for Software Teams | Atlassian." Atlassian. N.p., n.d. Web. 04 June 2016.