



Summer Fellowship Report

On

Implementation of Workflow, Searching, Sorting and Auto-Grader of LTI for eSim on Cloud

Submitted by

Rugved Somwanshi

Under the guidance of

Prof. Kannan Moudgalya

Chemical Engineering Department

IIT Bombay

April 2021

Acknowledgement

I, the summer intern of the **FOSSEE - eSim Cloud Project** am overwhelmed in all humbleness and gratefulness to acknowledge my deep gratitude to all those who have helped me put my ideas to perfection and have assigned tasks well above the level of simplicity and into something concrete and unique. I wholeheartedly thank **Prof. Kannan M. Moudgalya** for having faith in me, selecting me to be a part of his valuable project and for constantly motivating me to do better. I thank **Mr. Nagesh Karmali** and **Ms. Firuza Aibara** for providing me the opportunity to work on this project. I am also very thankful to my mentors for their valuable suggestions. They were and are always there to show me the right track when needed help. With help of their brilliant guidance and encouragement, I was able to complete my tasks properly and were up to the mark in all the tasks assigned. During the process, I got a chance to see the stronger side of my technical and nontechnical aspects and also strengthen my concepts. Last but not the least, I sincerely thank all my other colleagues working in different projects under **Prof. Kannan M. Moudgalya** for helping me evolve better with their critical advice.

Declaration

I declare that this written submission represents my ideas in my own words and whenever others' ideas or words have been included, I adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this thesis.

I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/-source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken when needed.

Rugved Somwanshi

Contents

1	Introduction	5
1.1	Problem Statement	5
1.2	Project Objective	5
1.3	Project Outcome	6
1.4	Project Requirements	7
2	Project Overview	8
2.1	Features	8
2.1.1	Workflow API	8
2.1.2	Searching and Sorting	11
2.1.3	Auto grader for LTI Apps	11
3	Feature Implementation	12
3.1	Workflow API	13
3.1.1	Updated Groups	13
3.1.2	States	15
3.1.3	Transitions	16
3.1.4	Projects	18
3.1.5	Transition History	20
3.1.6	Permissions	21
3.1.7	Reports	23
3.1.8	Miscellaneous	24
3.2	Searching and Sorting	25
3.3	LTI Grader	25
4	Conclusion	28
5	Future Work	29

List of Figures

3.1	Schema Diagram of Custom Group	13
3.2	Admin View of a Group	13
3.3	Schema Diagram of States	15
3.4	Admin Panel view of a state	15
3.5	Schema of Transition	16
3.6	Admin view of a Transition	17
3.7	Schema of Project	18
3.8	Create Project(1)	18
3.9	Create Project(2)	19
3.10	Project Page	19
3.11	Schema of Transition History	20
3.12	View of all logged transitions	20
3.13	Simple Project Timeline	21
3.14	Schema of Permissions	21
3.15	Admin view of a permission	22
3.16	Schema of Report	23
3.17	A user reporting a project	23
3.18	A reviewer approving a report	23
3.19	Reports shown to the contributor	24
3.20	Review Dialog for Reviewers	24
3.21	Review Dashboard for Reviewers	24
3.22	Sorted Schematic on the dashboard.	25
3.23	Searched Schematic on the dashboard.	25
3.24	Results if every parameter is wrong	26
3.25	Results if every parameter is correct	26
3.26	Results if some parameters are wrong	26
3.27	Graph Results	27

Chapter 1

Introduction

The current system allows the users to draw analog and digital circuits and simulate them. The users have a facility to drag and drop components from the left pane onto the schematic grid on the right pane. The components on the grid are connected using wires. The circuit can then be simulated using the different simulation parameters(DC Solver, DC Sweep, Transient analysis, and AC analysis). The basic ERC check enables the users to find out errors if any. The size of the schematic grid can be changed from A1 to A5 paper sizes along with portrait and landscape modes. The users can also print the circuit or save it in pdf format for documentation purposes.

Apart from this, contributors can create circuits and send them for review for publishing them on the platform for all to see.

All of this code is hosted in the repository available on [GitHub](#)

1.1 Problem Statement

The system had an absence of user roles and a common place for all users to see circuits. There was a need of a system for users to be able to publish created circuits and a review/reporting system for the same. This system should also support multiple roles and states of a circuit along side permissions for each role.

There was also a need of searching and sorting features for schematics on dashboard as it can easily get crowded and hamper user experience. Implementation of validation of circuits while simulating was required.

LTI integration for eSim on Cloud also required a way to calculate and send a grade to the LMS.

1.2 Project Objective

The objective of the project is to implement a workflow which enables a user to create *projects*, making the user a contributor, and having a multi-staged review process where a project goes through multiple states.

Searching and sorting based on different fields of a schematic is another objective of the project to make navigation to multiple circuits easier.

A basic auto-grader where comparison between simulation results is done to grade the submission is the final objective of the project.

1.3 Project Outcome

The projects add the facility to implement multiple states, roles and permissions for the workflow API which enables contributors to create new projects which will be reviewed and published on the site. The published projects are visible to all users and can be reported if any mistakes are present.

A search and sort functionality is implemented for Schematics, Projects and LTI apps on different pages,

Auto-grading with the aid of comparing the simulation results is done and the grade is submitted to the LMS.

1.4 Project Requirements

Following Major Technologies have been used during development.

- Docker
- Django (v2.2.12)
- React (v16.13.1)
- PostgreSQL

Chapter 2

Project Overview

The implemented features allow the user to create a project and contribute to the existing platform. A project consists of different details like procedure, observations and conclusion. This is not limited as the contributor can add his own custom details as he pleases. This project is sent for the reviewer to review and then is published after complete reviewing. This leads to a highly configurable workflow with evaluation checkpoints and a reporting system to revise the project if any mistakes are found by a user. A contributor can also provide default simulation parameters for a user to simulate. This system is also integrated to work with versioning.

A searching and sorting feature which searches over fields like title, author or description of a schematic, project or LTI app is also in place. Sorting takes place over multiple fields like name, creation time or save time is implemented. For reviewers, sorting based on name, author and status is also implemented.

An auto-grader is implemented by comparing the idea simulation results with simulation results submitted by a student over a LMS platform. A teacher, for example, can also set the parameters over which a submission shall be evaluated.

2.1 Features

My task was divided into three main parts, Workflow API, Searching/Sorting and Auto-grading Circuits. More details are given below.

2.1.1 Workflow API

In previous versions of eSim on Cloud, there was no place where all users can share their circuits in a common place, So with a goal to make the platform feel like a community and enable sharing different circuits and their observations over the cloud, an highly configurable workflow is created which enables users to be contributors by creating projects and publishing them to the platform after being reviewed by the reviewers.

This uses a role-based system where each user is assigned a role. Contributor role is given to every single user by default. Reviewer role can be given to certain users by the admin using Django admin panel. There is also an integration of staff roles. Every project has a state defining where it is currently in the workflow. The default

states are Draft, Review, Published and Reported.

Workflow

A default workflow is setup when the project is built. This workflow can be changed according to the use.

When a contributor creates a project, the project is in the *Draft* state. This is the stage where contributor can make changes, select the version of schematic it wants to use for the project and update the project if it has any changes.

After everything is in place, the contributor can put its project in *Review* State. This is the state where a reviewer can take up a project, review it and publish the project or send it back to draft if any mistake is found. The reviewer can also give a note to the contributor for pointing out necessary changes.

If everything is perfect, the project can be published for all users to see. The project goes into *Published* state. All published projects are visible in 'Projects' page.

A user can report a project if there are any issues. The project goes into *Reported* state. The reported project is still visible for other users but has a warning saying that the project is reported so not to take all details mentioned in the project as ground truth. A reviewer can then review the reports and approve or decline them if they are not correct. If the reports are approved, the reviewer must send this project back to *Draft* stage where the contributor can see the approved reports and fix the mistakes or make changes accordingly and send the project back to *Review* state.

Project page

A project consists of title, description, procedure, observation and conclusion. This can be modified by the contributor to its will. The project page also has an auto-generated component list, buttons to copy the schematic, simulate the given schematic in the project or report the project. A user has to give report description while reporting a project which helps the contributor to make proper changes. For simulation, the contributor can provide simulation parameters which will be auto-filled in the fields to aid users which insufficient knowledge in electronics. Any user can also view the timeline of the project, all the states the project went through and who reviewed the project. A user can also check the component parameters by double-clicking on the component present in the schematic.

Workflow Terminology

Here are a few parameters to keep in mind while configuring the workflow.

- States: The states which a project can be in. One can define if a project is a particular state, should it be visible to all the users or not.
- User Roles(Groups): Using Django Groups, one can define separate roles for both platforms, eSim and Arduino on cloud. There are attributes like is the role for staff or someone who should be reviewing the circuits.
- Permissions: This table is used to define view, edit and delete permissions for each state for each role. This enables granular control of permissions and can be extended in future to en-corporate more types permissions.
- Transitions: Very similar to permissions, this table is used to define which role is allowed to change a project in a specific state. This enables highly customized workflows for future use.

Integration with versioning

The project is integrated with versioning which allows contributor to try different circuits by making different variations and versions. Then, the contributor can select which specific version and variation of the circuit he wants to use for the project. This leads in easy and efficient revisions of a project and better user experience.

Miscellaneous

Whenever a user reports a circuit, a report is created which has to be accepted or denied by the reviewer. This helps in clearing wrong/malicious reports so the contributor can makes changes which are only necessary.

When a project is changed from one state to another, a *transition* takes place. This transition is logged in the back-end and can be viewed from the admin panel in the publish API side of things. This helps in establishing transparency to all the users about the flow itself and can be used for future reference if required.

2.1.2 Searching and Sorting

As schematics, projects and LTI apps go on increasing, there is a need of searching functionality which can help users find what they exactly want to access. There is also a sorting functionality implemented which can help users to sort their schematics for better user experience.

2.1.3 Auto grader for LTI Apps

A basic auto-grader is implemented which evaluates students simulation results by comparing it to teachers saved simulation when a student submits on the LMS platform. Depending the parameters defined by the teacher, the student is graded as well. This process is done in the back-end in a secure way. It also provides a feedback of which simulated parameters the student got correct. This could be worked used further to help student guide towards the right solution.

Chapter 3

Feature Implementation

The following chapter describes the procedure I had undertaken to implemented the above listed features:

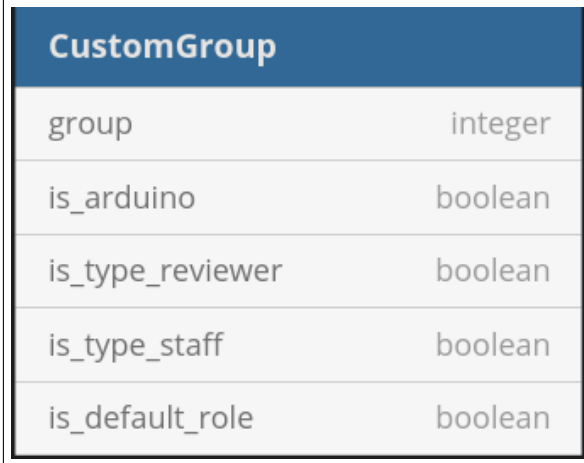
- 1) Workflow API
- 2) Searching and Sorting
- 3) Auto grader for LTI Apps

3.1 Workflow API

Implementing workflow came with adding models like Projects, States, Transitions, Transition History, Reports and Permissions. Groups model was updated as well. Their implementation is given below:

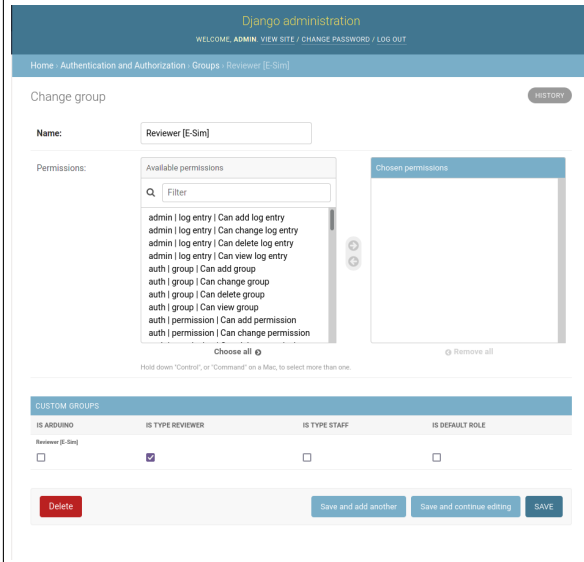
3.1.1 Updated Groups

This model is used to extend the functionality of the existing groups system in Django. These groups are used as roles in the workflow.



CustomGroup	
group	integer
is_arduino	boolean
is_type_reviewer	boolean
is_type_staff	boolean
is_default_role	boolean

Figure 3.1: Schema Diagram of Custom Group



Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home / Authentication and Authorization / Groups / Reviewer [E-Sim]

Change group HISTORY

Name:

Permissions:

Available permissions

Filter

- admin | log entry | Can add log entry
- admin | log entry | Can change log entry
- admin | log entry | Can delete log entry
- admin | log entry | Can view log entry
- auth | group | Can add group
- auth | group | Can change group
- auth | group | Can delete group
- auth | group | Can view group
- auth | permission | Can add permission
- auth | permission | Can change permission

Choose all

Hold down "Control", or "Command" on a Mac, to select more than one.

Chosen permissions

Remove all

CUSTOM GROUPS

IS ARDUINO	IS TYPE REVIEWER	IS TYPE STAFF	IS DEFAULT ROLE
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Reviewer [E-Sim]

Delete Save and add another Save and continue editing SAVE

Figure 3.2: Admin View of a Group

The attributes are:

- `group`: Key of the Django group it is referencing.
- `is_arduino`: If the specific role/group is for Arduino or for eSim on cloud.
- `is_type_reviewer`: If the specific role/group is for reviewers. This grants the privilege to review reports for that specific role.
- `is_type_staff`: If the specific role/group is for staff. This grants the privilege to change gallery circuits for that specific role.
- `is_default_role`: If the specific role/group is supposed to assigned to every new user created on the platform.

Default groups are:

- Contributor[eSim]
- Contributor[Arduino]
- Reviewer[eSim]
- Reviewer[Arduino]
- Staff[eSim]
- Staff[Arduino]

3.1.2 States

Every Project will have a state it is currently in. These states define where a particular project is in the current workflow.

State	
id	int
name	varchar
description	varchar
public	boolean
report	boolean

Figure 3.3: Schema Diagram of States

The screenshot shows the Django administration interface for editing a state. The page title is 'Django administration' with a welcome message for 'ADMIN'. The breadcrumb trail is 'Home > Workflow API > States > Review'. The main heading is 'Change state' with a 'HISTORY' button. The form contains the following fields:

- Name:** A text input field containing 'Review'.
- Description:** A text input field containing 'State used for the reviewing stage'.
- Public:** A checkbox that is currently unchecked.
- Report:** A checkbox that is currently unchecked.

At the bottom of the form, there are four buttons: 'Delete' (red), 'Save and add another' (light blue), 'Save and continue editing' (light blue), and 'SAVE' (dark blue).

Figure 3.4: Admin Panel view of a state

The attributes are:

- name: Name of the state

- description: Description of the state
- public: A boolean value to define if a project is in this particular state, should it be visible to any user on the platform.
- report: A boolean value to define if a project is in this particular state, should it be considered reported and sent for reviewers to review. Note that this is only used for the reporting side (to enable controlling of reports) of the workflow and should not be ticked for the normal Review state.

Default States are:

- Draft: State used for the drafting stage of a project
- Review: State used for the reviewing stage
- Reported: State used when a project gets reported
- Published: State used for the published stage

3.1.3 Transitions

Each Transition defines which roles can change the state of a project from state A to state B. It also holds the messages for each event and history to display it on a timeline. History is defined as an action which was taken on a project and an event is defined as something which is currently happening. Each type of user will have different messages shown. These details are then visible in the timeline of a project.

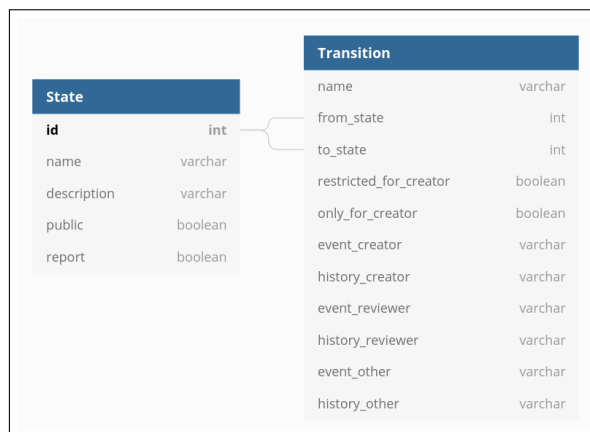


Figure 3.5: Schema of Transition

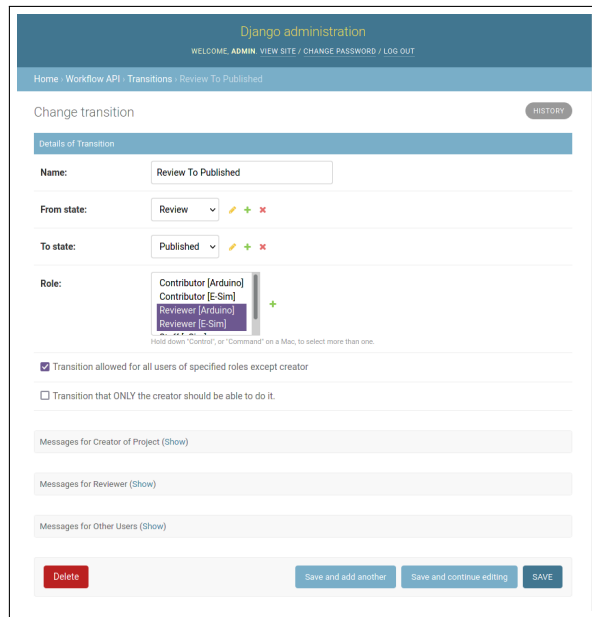


Figure 3.6: Admin view of a Transition

The attributes are:

- **name:** Name of the transition.
- **from_state:** The state in which the project is expected to be in.
- **to_state:** The state to which the project is expected to be put in.
- **restricted_for_creator:** If the transition is restricted for creator. If True, the creator of the project cannot make this transition happen.
- **only_for_creator:** If the transition is allowed ONLY for creator. If True, the creator of the project only can make this transition happen.
- **event_creator:** Event message for the creator of the project.
- **history_creator:** History message for the creator of the project.
- **event_reviewer:** Event message for a reviewer.
- **history_reviewer:** History message for a reviewer.
- **event_other:** Event message for a normal user.
- **history_other:** History message for a normal user.

3.1.4 Projects

The entity which goes from one state through another and for which the workflow is established. The contributor can specify different types of details for the project, which is fully customizable, and also define the default simulation parameters. The contributor also has to select the version of the schematic which needs to be used for the project.

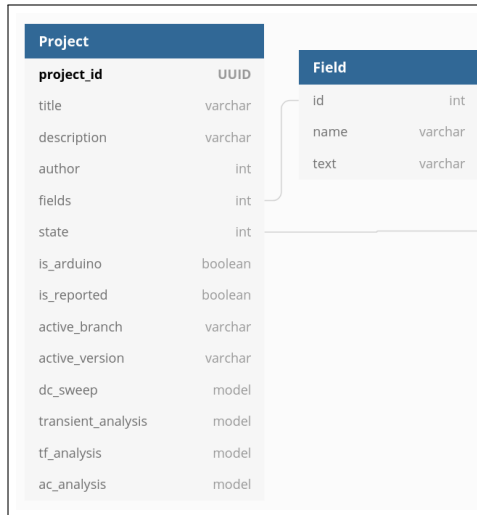


Figure 3.7: Schema of Project

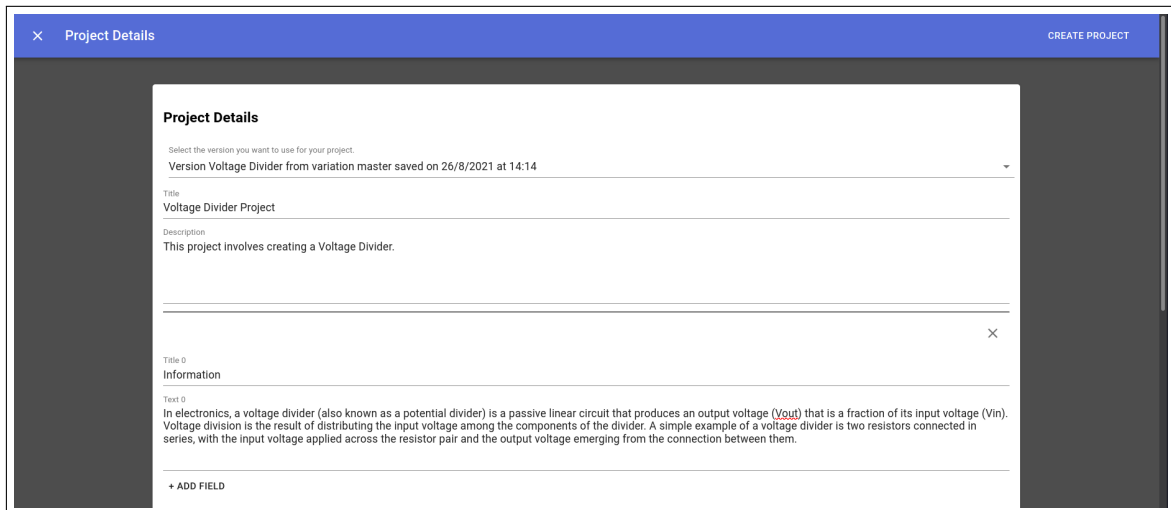


Figure 3.8: Create Project(1)

Simulation Parameters

Select simulation mode parameters to enter:
DC Sweep

DC Sweep

Select Component V1

Start Voltage V

Stop Voltage V

Step V

Secondary Parameters

Select Component

Start Value

Stop Value

Step Value

Figure 3.9: Create Project(2)

eSim
HOME EDITOR GALLERY PROJECTS SIMULATOR DASHBOARD

Voltage Divider Project

By: admin

This project involves creating a Voltage Divider.

Component List:

- 1.V1 1V
- 2.R1 1kOhm
- 3.R2 2kOhm

Information:

In electronics, a voltage divider (also known as a potential divider) is a passive linear circuit that produces an output voltage (V_{out}) that is a fraction of its input voltage (V_{in}). Voltage division is the result of distributing the input voltage among the components of the divider.

Circuit Diagram:

GENERATE NETLIST
SIMULATE
MAKE A COPY
REPORT

🔍

🔍

📄

Figure 3.10: Project Page

3.1.5 Transition History

This model is responsible for logging all transitions taking place on the platform. This also aides in displaying a timeline on the Project page and create project page for reference of the users.

TransitionHistory	
id	integer
project	Project
transition	Transition
transition_author	User
transition_time	timestamp
reviewer_notes	varchar
is_done_by_reviewer	boolean

Figure 3.11: Schema of Transition History

The screenshot shows the Django administration interface for 'Transition Histories'. At the top, it says 'Django administration' and 'WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT'. Below that, the breadcrumb 'Home > Publishapi > Transition Histories' is visible. The main content area has the heading 'Select transition history to change' and an 'ADD TRANSITION HISTORY +' button. There is an 'Action:' dropdown menu and a 'Go' button, with '0 of 2 selected' indicated. A table lists two transition histories:

<input type="checkbox"/>	ID	TRANSITION	TRANSITION AUTHOR	TRANSITION TIME
<input type="checkbox"/>	2	Review to Draft	admin	Aug. 26, 2021, 2:19 p.m.
<input type="checkbox"/>	1	Draft To Review	admin	Aug. 26, 2021, 2:18 p.m.

Below the table, it says '2 Transition Histories'.

Figure 3.12: View of all logged transitions

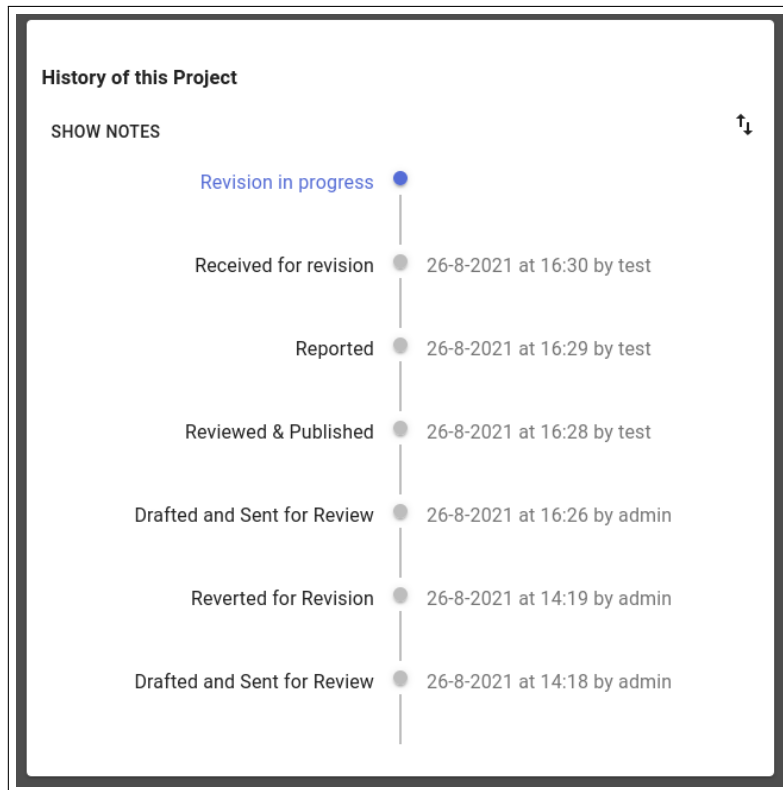


Figure 3.13: Simple Project Timeline

3.1.6 Permissions

This model is responsible for configuring View, Edit and Delete permissions for each role in reference to Projects.

Permission	
int	id
role	Group
view_own_states	State
edit_own_states	State
view_other_states	State
del_own_states	State

Figure 3.14: Schema of Permissions

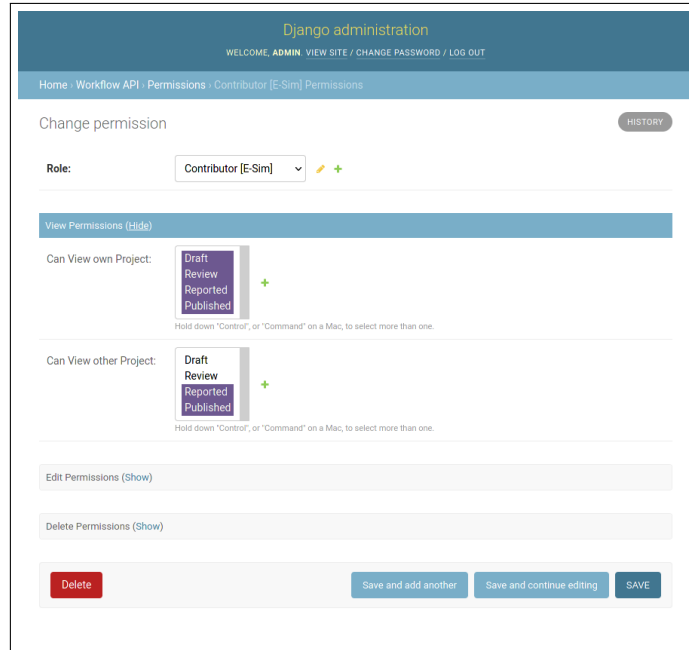


Figure 3.15: Admin view of a permission

The attributes are:

- `role`: Group/Role to which this permission is associated.
- `view_own_states`: Permission to view your own project in defined states.
- `edit_own_states`: Permission to edit your own project in defined states.
- `view_others_states`: Permission to view others project in defined states.
- `del_own_states`: Permission to delete your own project in defined states.

3.1.7 Reports

This model is used for implementing a report system for the project. When a user reports a project, the report is sent to the reviewer. The reviewer check the legitimacy of the report and approves it likewise. If the report is approved, it is sent to the contributor of the project so for the contributor to make the necessary changes.

Report	
id	int
project	project
report_open	boolean
resolver	User
report_time	timestamp
description	varchar
reporter	User
approved	boolean

Figure 3.16: Schema of Report

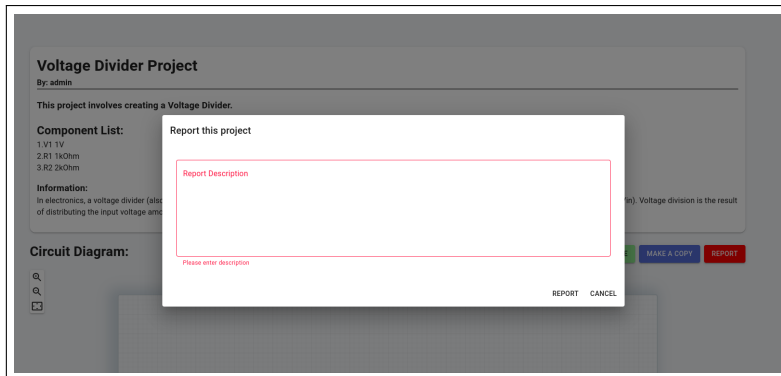


Figure 3.17: A user reporting a project

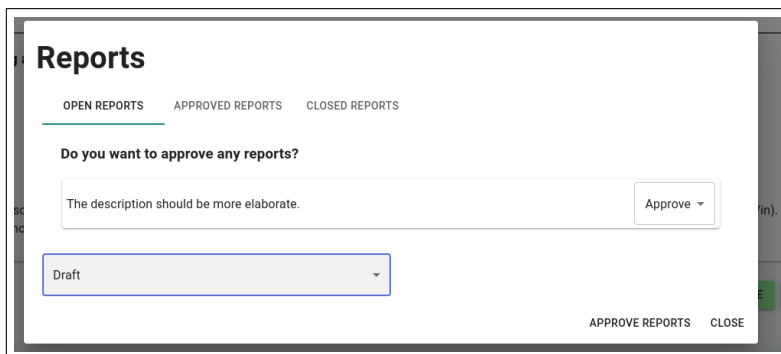


Figure 3.18: A reviewer approving a report

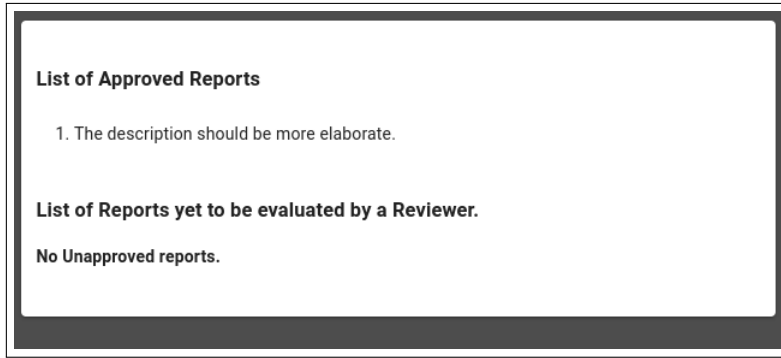


Figure 3.19: Reports shown to the contributor

3.1.8 Miscellaneous

While reviewing, a reviewer can put notes so for any user to see what the reviewer has in mind. This can be used to let contributor know any changes if the project is sent back to the Drafting state.



Figure 3.20: Review Dialog for Reviewers

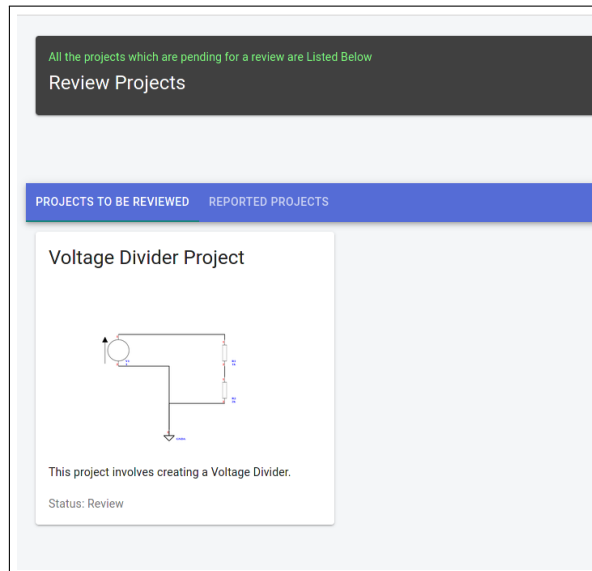


Figure 3.21: Review Dashboard for Reviewers

3.2 Searching and Sorting

A searching and sorting feature is implemented in the dashboard of eSim on cloud. This helps in easy navigation between multiple schematics, projects or LTI apps and submissions.

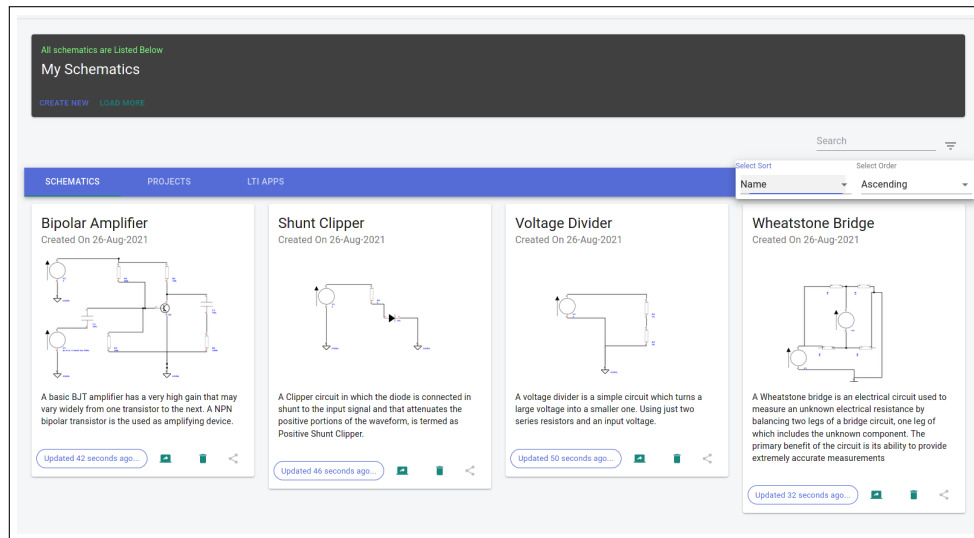


Figure 3.22: Sorted Schematic on the dashboard.

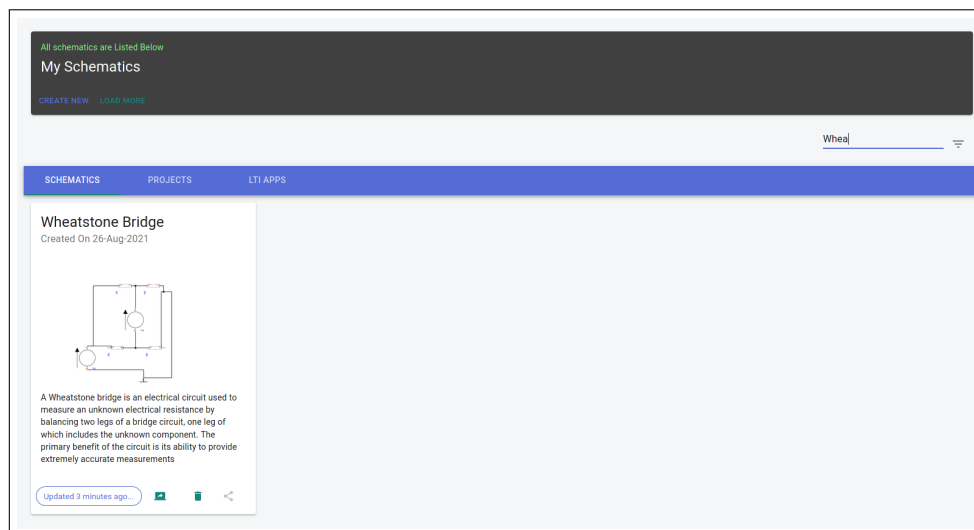


Figure 3.23: Searched Schematic on the dashboard.

3.3 LTI Grader

Implemented a basic grader which compares teacher and students simulation results to give a grade. Each simulation parameter is compared and evaluated accordingly.

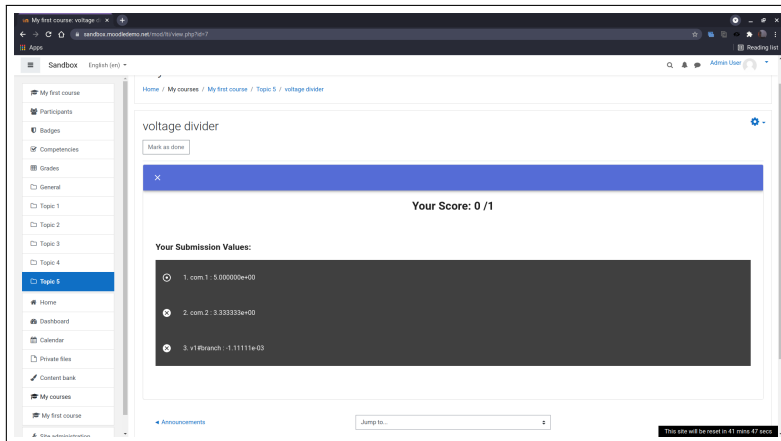


Figure 3.24: Results if every parameter is wrong

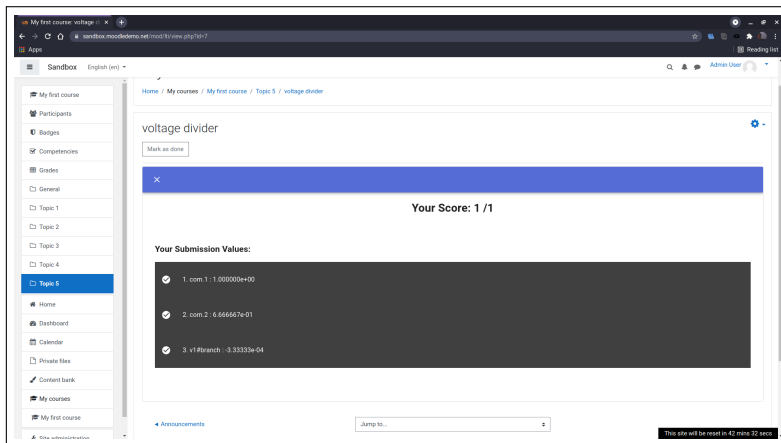


Figure 3.25: Results if every parameter is correct

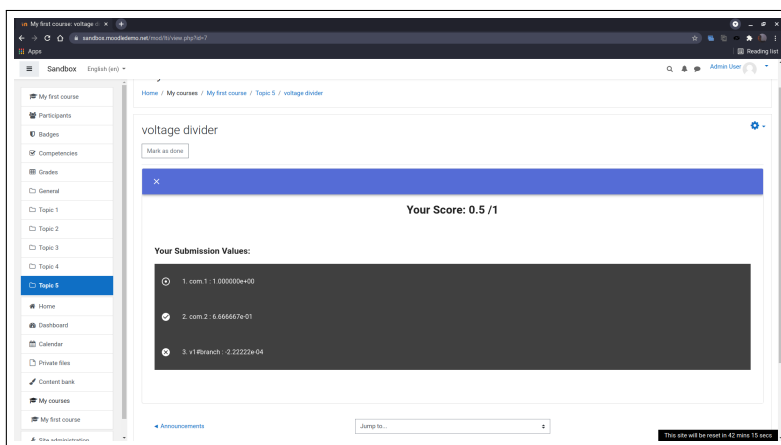


Figure 3.26: Results if some parameters are wrong

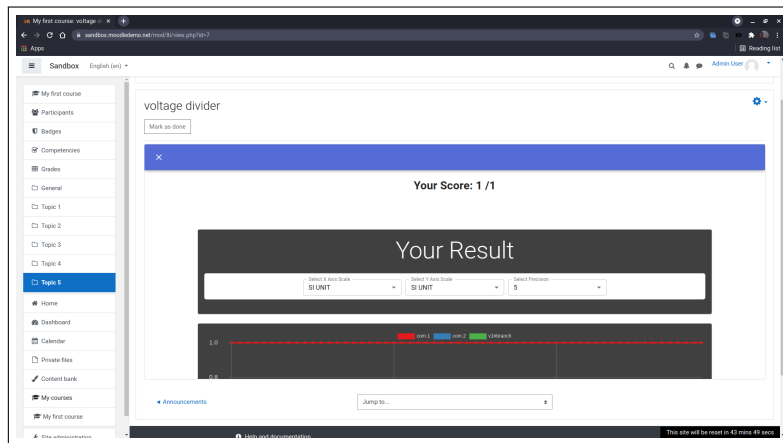


Figure 3.27: Graph Results

Chapter 4

Conclusion

In conclusion, I have successfully implemented Workflow, Searching/Sorting and LTI Grade.

The workflow will aid in making the platform more cohesive and establishing a community type system which will help all users exchange different schematics and information. This will encourage more users to contribute and use the platform likewise.

Searching and sorting helps uplift the user experience as a lot of new types of features like projects and LTI apps are introduced. This helps in easier navigation and future-proofing navigation in general. LTI grader implementation helps the LTI system up and running. This can be used as a foundation to improve the grader further along and implement different types of evaluations.

This project has helped me a lot of new technologies, good practices and designing a robust system for many users to use.

Chapter 5

Future Work

For the workflow API, there is a need of implementing comments and a common feed for Projects. This will give the platform a sense of community and will make users want to come back to the site. The permissions section can be extended to implement all different types of permissions for the platform, expanding it just from permissions for projects. There can be a metric system which can show which contributor contributes more and which reviewer reviews more. This can result into an incentive to have more engaging activity on the platform. A upvote/downvote system just like Reddit for projects can be implemented too. For the LTI grader, different ways to improve grading such as weighted grading and partial checking of results can be done.

Bibliography

- [1] <http://ngspice.sourceforge.net/docs/ngspice-manual.pdf> (Accessed on April 10, 2021).
- [2] <https://www.django-rest-framework.org/> (Accessed on April 10, 2021).
- [3] <https://www.djangoproject.com/> (Accessed on April 10, 2021).
- [4] <https://swagger.io/> (Accessed on April 20, 2021).
- [5] <https://reactjs.org/> (Accessed on April 20, 2021).