

03/17/2022



Developer Student Clubs

University of Toronto Mississauga

GDSC2.0

System Design Document

**Stephan Motha, Vishay Singh, Sahib Nanda, Dale Rodrigues, Litao
Chen, Ahmed Al-Mandalawi**

Table of Contents

- 1. CRC cards**
- 2. Software architecture diagram**

CRC Cards

CRC Maker

2022-03-17, 7:39 PM

Abstract	Models Application, Log, Admin, Projects, Resources, Team Models	None
<ul style="list-style-type: none">• Define Schemas for mongoDB	<ul style="list-style-type: none">• Databases	

	Databases Application, Log, Admin, Projects, Resources, Team Schemas	None
<ul style="list-style-type: none">• Store Data• Query Data• Create Schema	<ul style="list-style-type: none">• Models• Controller	

	Config (environment)	None
<ul style="list-style-type: none">• Define environment variables (proxy port and database connection)	<ul style="list-style-type: none">• None	None

	Server	None
<ul style="list-style-type: none">• Provide communication instance for endpoints and frontend• Store application global variables and states	<ul style="list-style-type: none">• Axios	Routes, Controllerss

	Routes	Server
<ul style="list-style-type: none">• Define paths for endpoints	<ul style="list-style-type: none">• Controller• Endpoints	Endpoints

Controller Application, Team and Resources Endpoints		Server
<ul style="list-style-type: none">• Define the functionality of endpoints• Handle requests and response• Query databases	<ul style="list-style-type: none">• Databases• Routes	

Axios		None
		None
<ul style="list-style-type: none">• Sending requests to Server• Receiving response from Server	<ul style="list-style-type: none">• Components• Server	

Components (App) Login, Register, Application, Resource, Projects, Admin Interface, General Info Views		None
<ul style="list-style-type: none">• Define URLs for pages• Render pages• Validate Payloads	<ul style="list-style-type: none">• Axios	

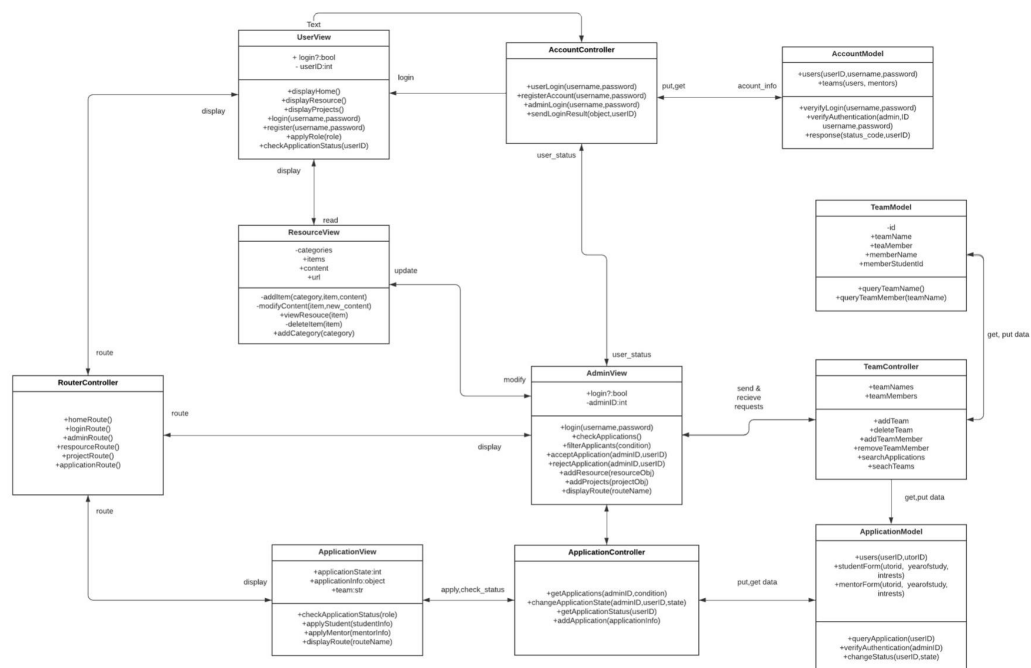
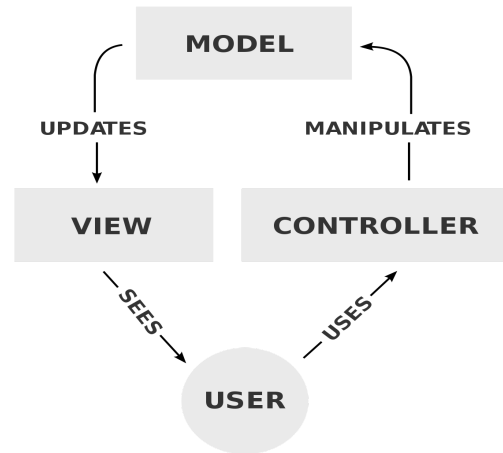
Modifications from Sprint2 Doc:

- Models: update team models
- Databases: specify the schemas we have
- Components: update newly added frontend

Software Architecture Diagram

The software architecture design our team chooses is the MVC or (model-view-controller) design as it complimented the way our team wanted to design and develop our system. The model handles the logic and data of the software. The view is responsible for the display and user interface. The controller is the connection that processes information to and from the model and view.

In our software, we have a frontend folder that corresponds to the view, a backend folder that corresponds to the model, and a file called server.js that represents the controller. The frontend contains React components that interact with each other and the controller to create an interactive user experience. The backend contains models that represent how the data should look in the database. It also contains routes that perform a specific action in the backend based on instructions from the controller. These instructions are usually in the form of “get” or “post” requests. The controller server.js receives instructions from the frontend, these usually include endpoints, and decides which route to call in the backend.



Modifications from Sprint2:

- Adding Team Controller and Model

For a better view, please view the original pdf file in our doc.