

03/03/2022



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

2022-03-03, 9:49 PM

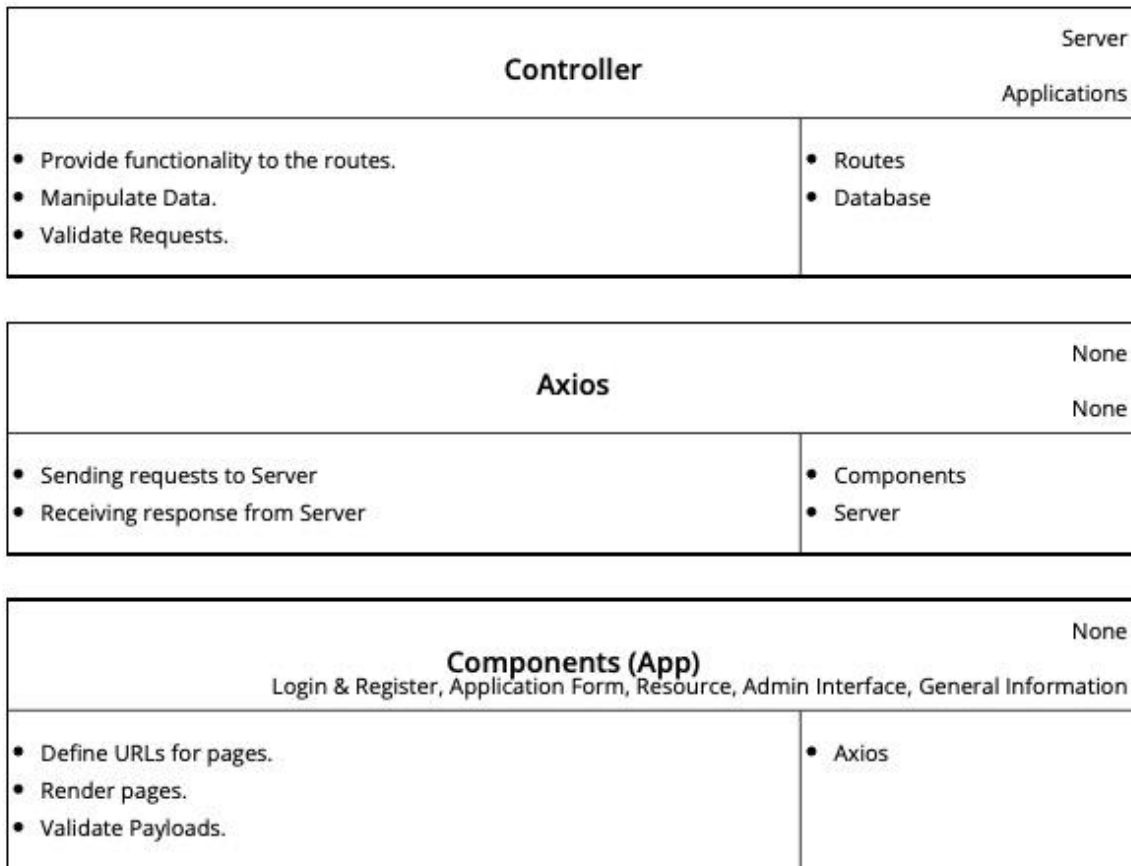
Models mentorAppModel, studentAppModel, teams, testModel, users		None
• Define Schemas for MongoDB	• Databases	

Databases		None
		None
• Store Data • Query Data • Create Schema	• Models • Controller	

Config (environment)		None
		None
• Define environment variables (server port and database connection)	• None	

Server		None
		Routes, Controller
• Provide communication instance for endpoints and frontend. • Define global variables or states which need to be used in the entire application.	• Axios	

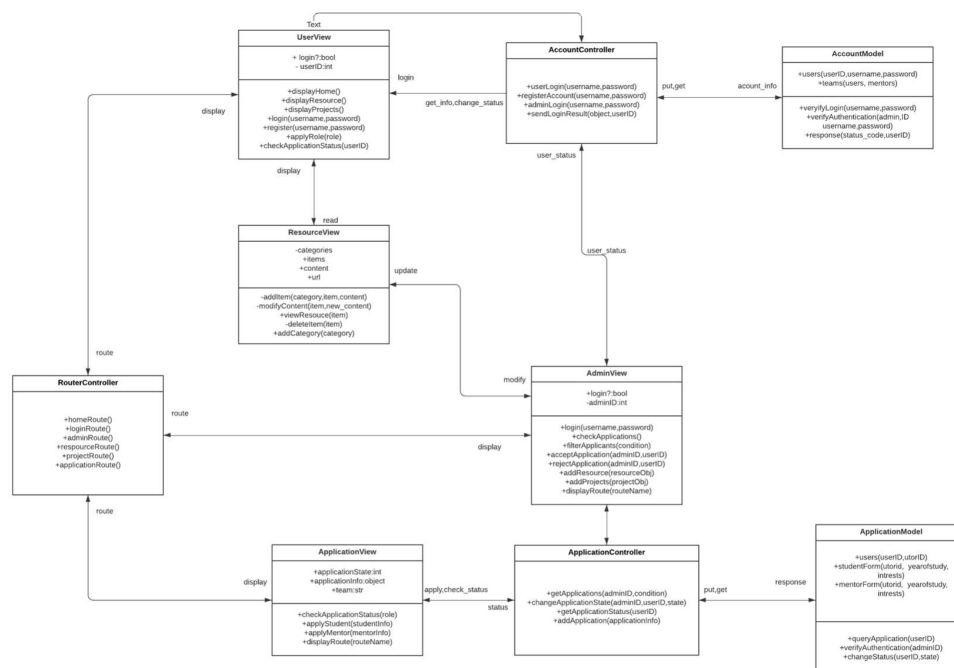
Routes		Server
		applications, users
• Provide endpoint paths	• Controller	



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

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.



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