

VIP/Senior Project Final Presentation

Spring 2019

INTERACTIVE SMART TABLE

Team Member(s): William Escudero
Product Owner(s): Hadi Alhaffar
Professor: Masoud Sadjadi

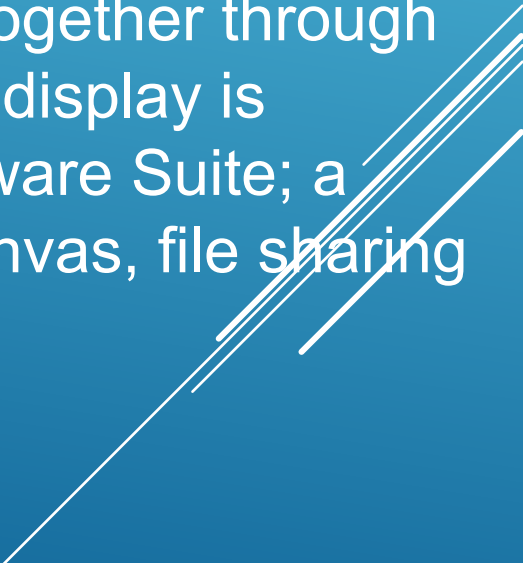
School of Computing and Information Sciences
Florida International University

April 18, 2019

PROBLEM DEFINITION

As technology takes over meeting and conference rooms, the desk remains unchanged and offline. This can reduce interaction and engagement which can decrease productivity and lower the effectiveness of the event.

Interactive Smart Table (IST) is a table for the conference or meeting room that connects every member together through a smart display on every desk sit. The smart display is powered by the Interactive Smart Table Software Suite; a collaboration suite that provides a shared canvas, file sharing and slide presentation capabilities.



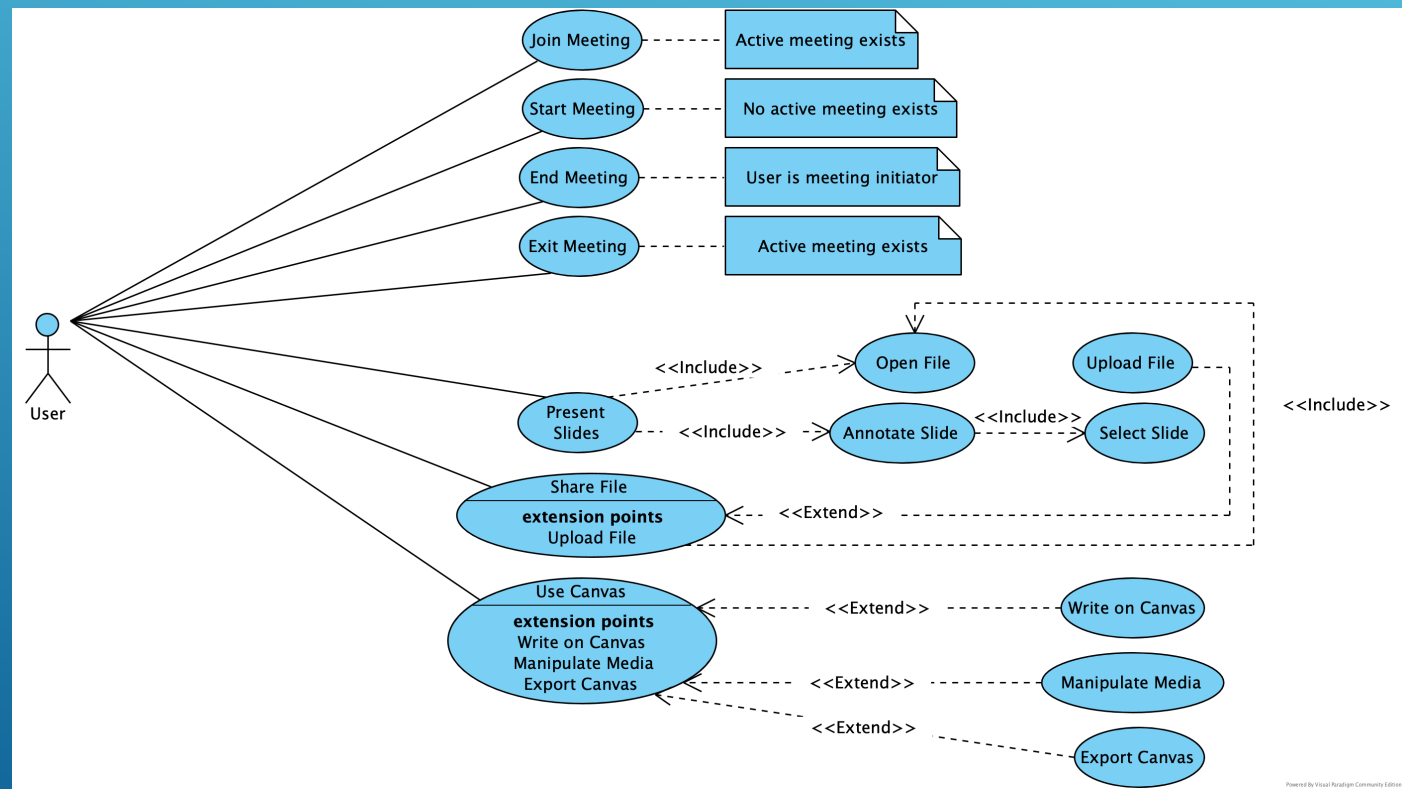
REQUIREMENTS: USER STORIES

Story	Description
INTR-60	Research if input should come from screen or external device.
INTR-61	Research screens that will integrate well with raspberry pi and support input method.
INTR-62	Integrate screen with raspberry pi.
INTR-63	Go over documentation for interactive smart table software built by previous team.
INTR-64	Installation of previous software on hardware platform.
INTR-70	As a user I would like to have the option to write on a canvas.
INTR-71	UI Redesign to be more modular so more features are included
INTR-72	As a user I want to create a meeting and invite meeting members so that I can present and share documents.
INTR-76	As meeting initiator I would like to end a meeting
INTR-77	As a user I would like to have a component that will allow me to brainstorm by putting different types of content so that I can more easily put a meeting or presentation together
INTR-78	As a user I want to be able to clear the canvas so that I can start anew.
INTR-79	As a user I want to be able to group multiple items into a single item so that I can manipulate them as a unit.
INTR-81	As a user I want to be able to have a shared canvas so that every action done on said canvas can be reflected remotely to other meeting participants and vice versa.

Issues 72, 77 and 81 are the most important as they create a working system.

REQUIREMENTS: USE CASES

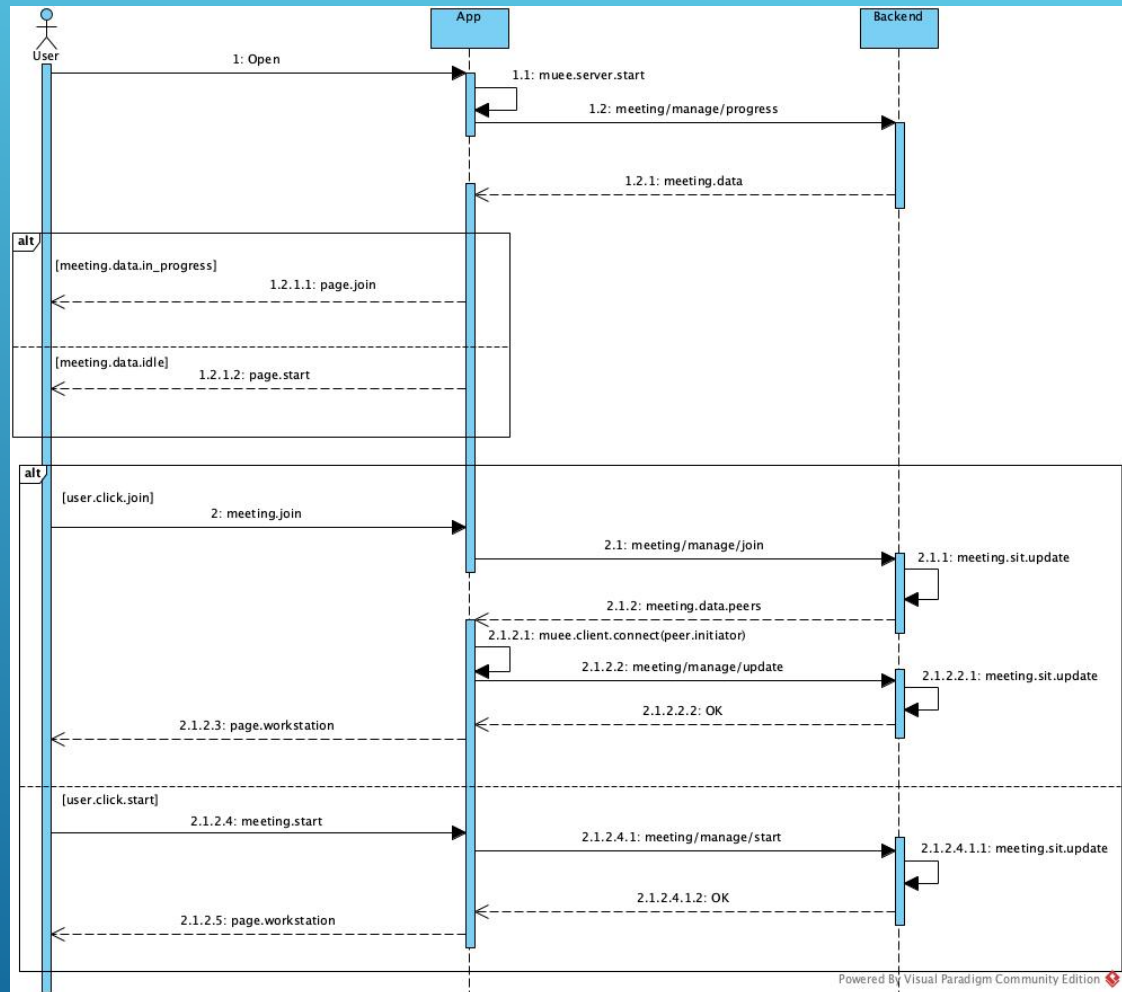
The general use case below describes the most important aspects of the system and how different objects interact together



REQUIREMENTS: SEQUENCE DIAGRAMS

The sequence diagram for meeting management models how the meetings get started or how users join in.

This sequence diagram showcases how the application interacts with the backend system.

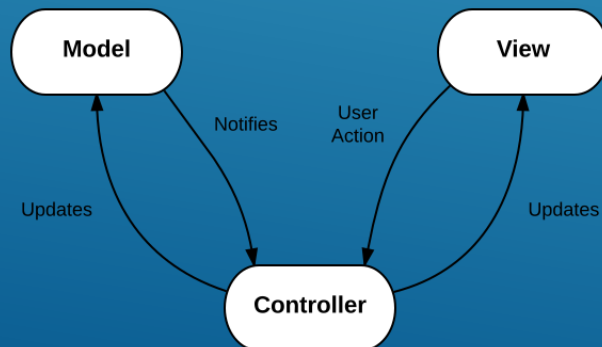


SYSTEM DESIGN: ARCHITECTURE

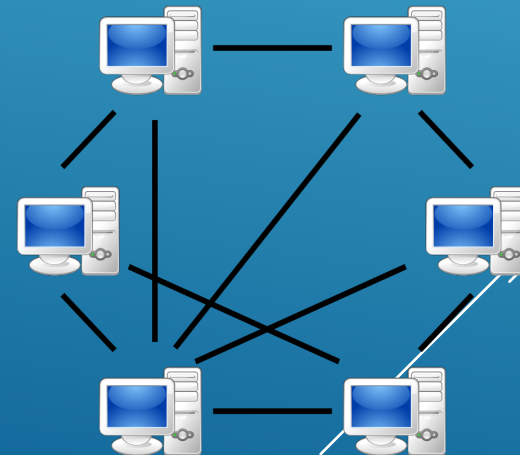
The Interactive Smart Table Software Suite (ISTSS) is composed of two major subsystems; The backend and the application.

The MVC architectural pattern was used for the backend while the Event Driven and Hybrid p2p architectural patterns were used for the application.

MVC



P2P



SYSTEM DESIGN: ARCHITECTURE

Application

The application uses three subsystems working together to achieve its goal. The network, backend and event subsystems are combined to create MUEE, Multi User Event Engine.

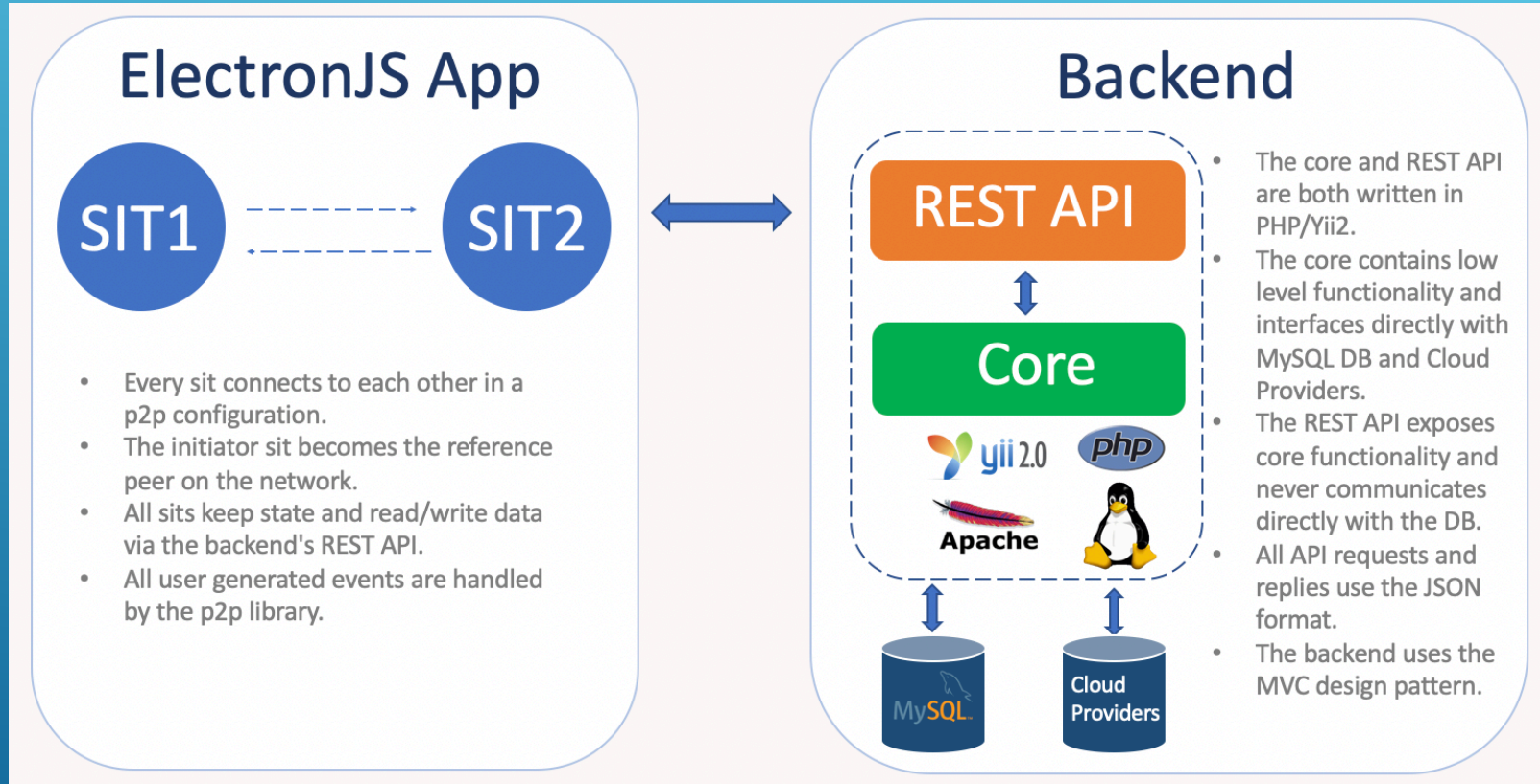
- **Backend.js**: This component handles all the communication between the application and the backend. It contains all the requests to the backend's API and it is used by the Network and Event subsystem in order to update states, get statuses, etc.
- **Network.js**: This component implements the p2p functionality for the application. It creates a TCP client and a TCP server when the application starts and starts to listen for peer connections. Network.js also contains the implementation of the MUEE Protocol which is fundamental and it is used to encode actions into events and sent over the network. The protocol is used by Event.js
- **Event.js**: This library takes any action generated within the application and encodes it into an event using the MUEE protocol. For instance when a user draws on canvas a representation of that action using MUEE protocol can be expressed as
<user_action>:<action_type>:<action_data>:<delimiter>

Backend

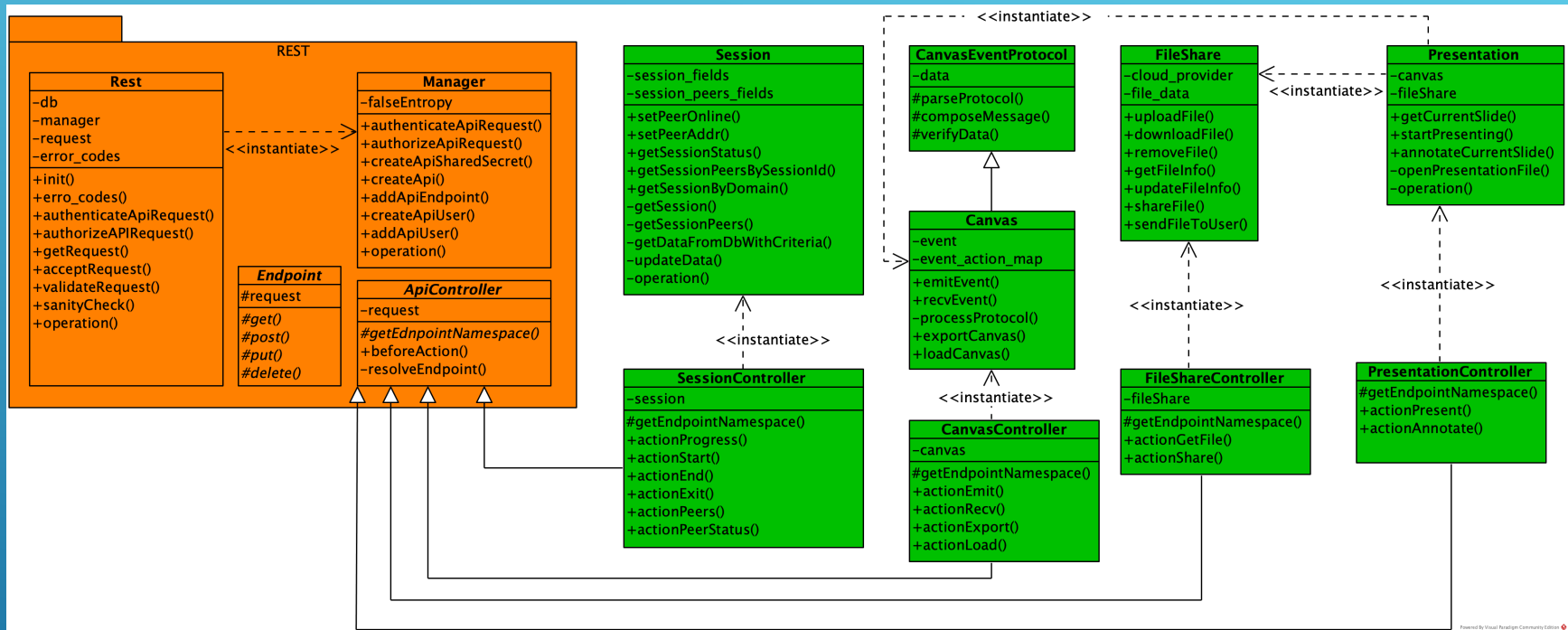
The backend is the data processing subsystem and it exposes the core functionality through a REST API.

- The application sends a request to the backend and the backend replies with some data. The application then uses this data to update whatever it needs to in terms of UI components or states.
- The backend does not keep any states (stateless). The backend also functions as the p2p index server (super peer)

SYSTEM DESIGN: DEPLOYMENT



HIGH LEVEL CLASS DIAGRAM



The high level class diagram gives an overview of the overall architecture of the system. The classes in orange symbolize the REST subcomponent while the green symbolizes the application Core.

SUMMARY

Increased meeting Interaction and engagement by connecting all meeting members via the Interactive Smart Table Software Suite.

File sharing and Canvas component allow the user to contribute in real time while in a meeting.

CURRENT SYSTEM

High portability and design efficiency allows the software suite to run on any platform that supports ElectronJS, decoupling the table design from the software that powers it.

The backend subsystem is running on a Google Cloud VM instance and is fully operational.

The application runs on prototype platform composed of a Raspberry PI 3 Model B+ connected to a Raspberry PI 7" touch screen.

CONTACT

William Escudero

wescu001@fiu.edu

<http://www.williamescudero.com>