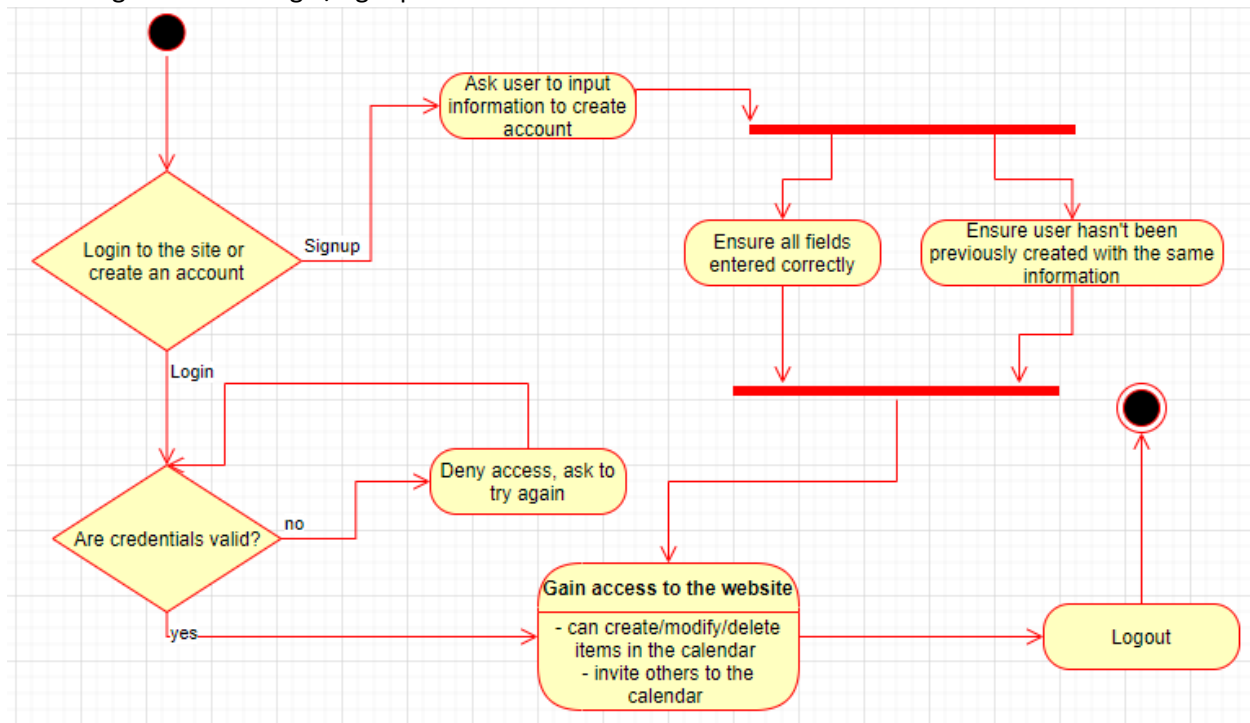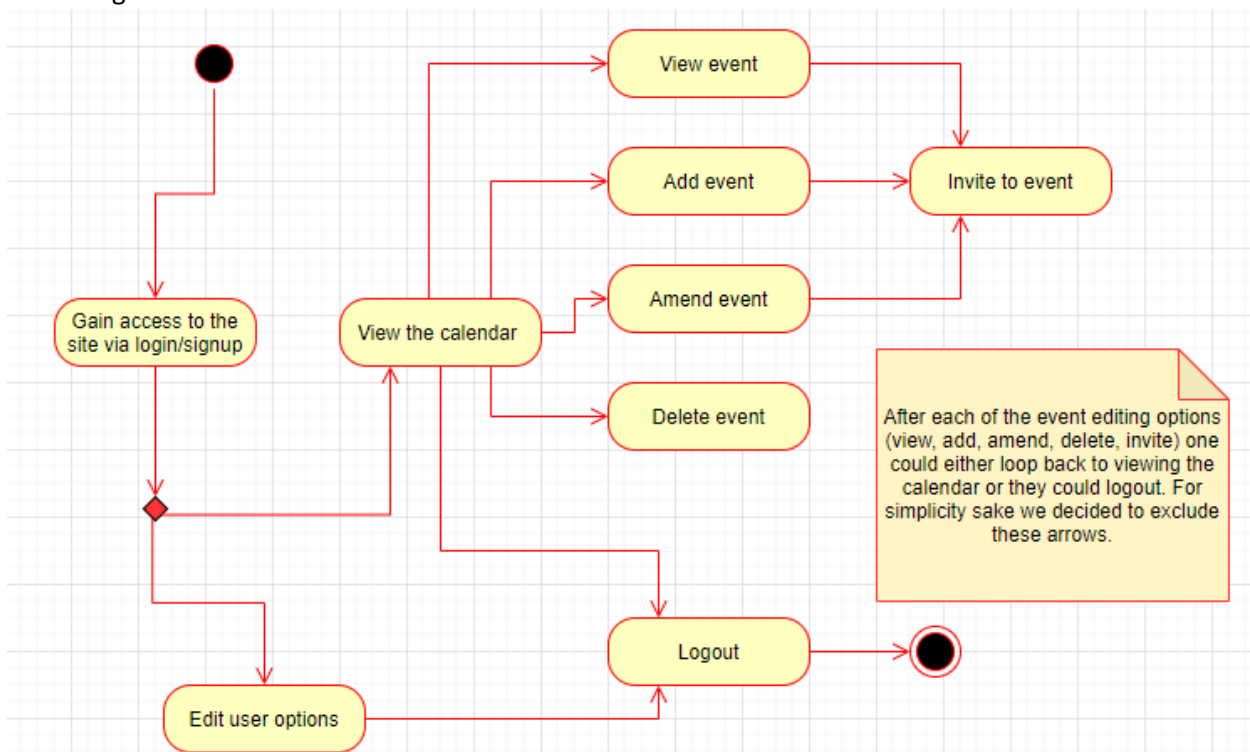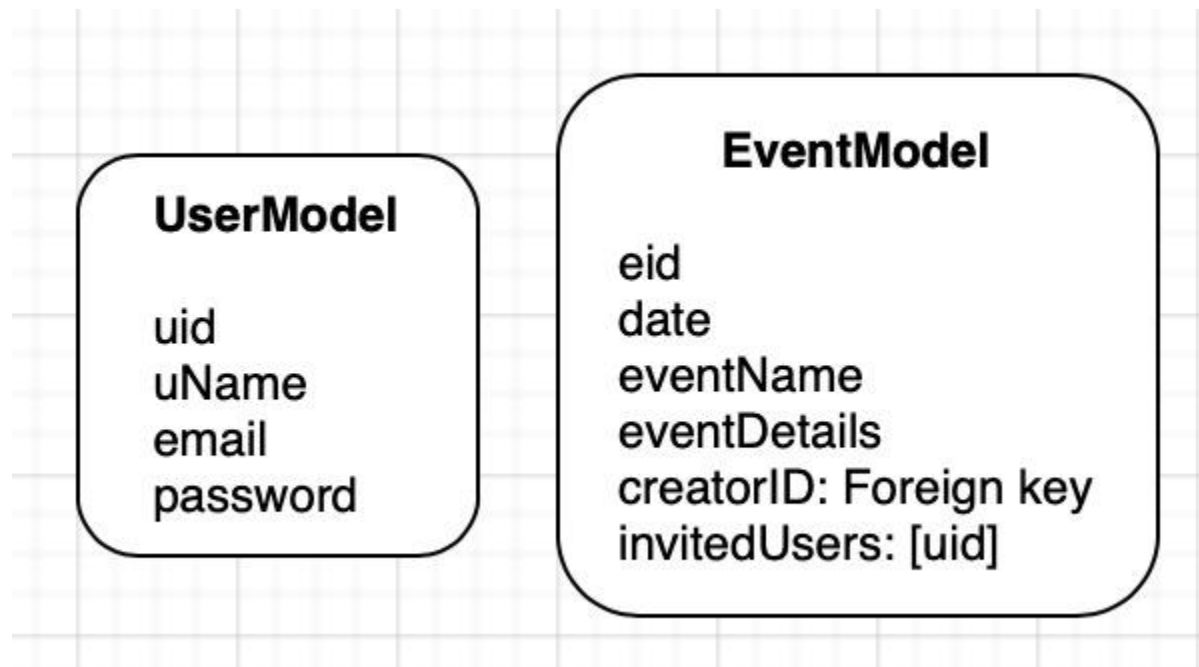State diagram for the login/signup state:
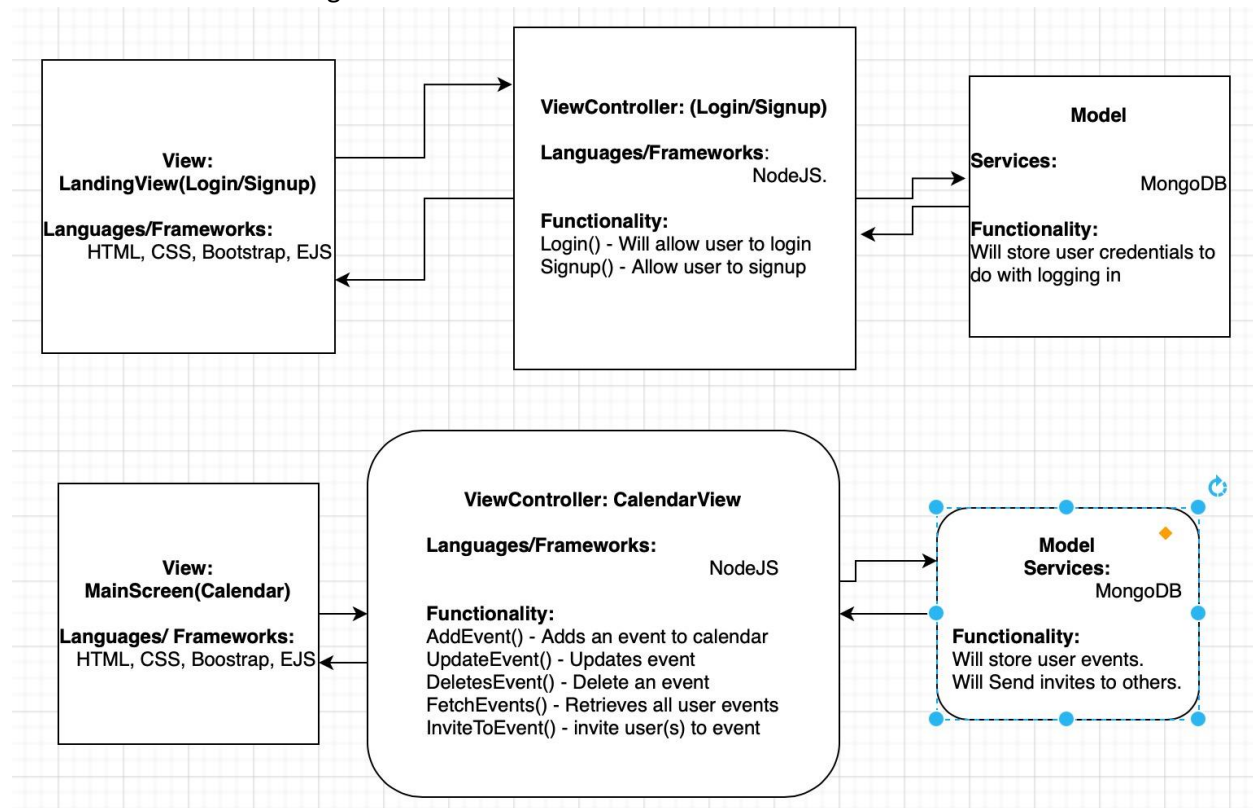

State diagram for the Website Traversal State:

Data Model Diagram:

UserModel

uid
uName
email
password

EventModel

eid
date
eventName
eventDetails
creatorID: Foreign key
invitedUsers: [uid]

Model View Controller Diagram:

View:
LandingView(Login/Signup)

Languages/Frameworks:
HTML, CSS, Bootstrap, EJS

ViewController: (Login/Signup)

Languages/Frameworks:
NodeJS.

Functionality:
Login() - Will allow user to login
Signup() - Allow user to signup

Model

Services:
MongoDB

Functionality:
Will store user credentials to
do with logging in

View:
MainScreen(Calendar)

Languages/ Frameworks:
HTML, CSS, Boostrap, EJS

ViewController: CalendarView

Languages/Frameworks:
NodeJS

Functionality:
AddEvent() - Adds an event to calendar
UpdateEvent() - Updates event
DeletesEvent() - Delete an event
FetchEvents() - Retrieves all user events
InviteToEvent() - invite user(s) to event

Model
Services:
MongoDB

Functionality:
Will store user events.
Will Send invites to others.

Class Diagram:

LoginSignupViewController

Login()
Signup()

* Calls AuthService functions async
and handles errors

AuthService

Login()
Signup()
Logout()

* Calls DB directly

CalendarViewController

AddEvent()
UpdateEvent()
DeleteEvent()
FetchEvents()
InviteToEvent()

* Calls EventService functions
async and handles errors

EventService

AddEvent()
UpdateEvent()
DeleteEvent()
FetchEvents()
InviteToEvent()

* Calls DB directly

Lofi design of the login/signup page, the calendar view, editing an event and how the menus will look: