

Student: Alvaro Villa, Florida International University
Mentor: Annette King, Citrix
Instructor: Masoud Sadjadi, Florida International University

Special thanks to: Annette King, Kyle Roque, and Alice Kwok

Senior Project – Fall 2017

Purpose

The system aims to help cancer patients, their caregivers, and healthcare providers by serving up helpful and timely content and resources related to a patient's diagnosis and profile to ease their treatment journey.

Current System

- **Create and edit profile** with the option of remaining anonymous as per HIPAA compliance laws.
- **Engage in the community** by posting to forums or joining/creating events.
- **View resources** and provide them if user has admin/provider status.
- **Delete documents** as admin from database collections.
- **Schedule appointments** and/or medications for patients.

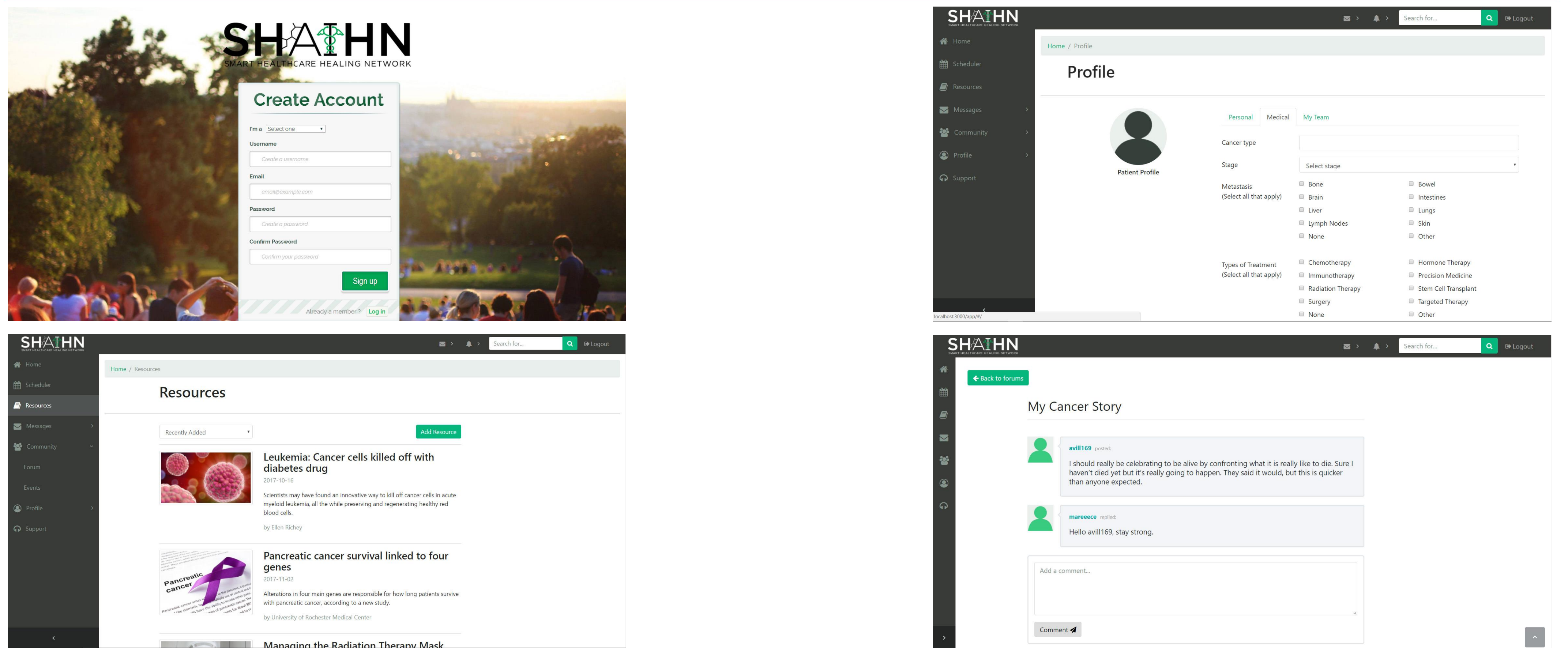
Requirements & Verification

- **Community resource dashboard** for caregivers, patients, and providers.
- **Front-end web application** to facilitate care plan dissemination, tracking and monitoring communications and caregiver support.
- **Content management** for tailored patient and caregiver support.
- **Network feature** to connect with peer support groups.
- **Manual testing** was used to verify system.

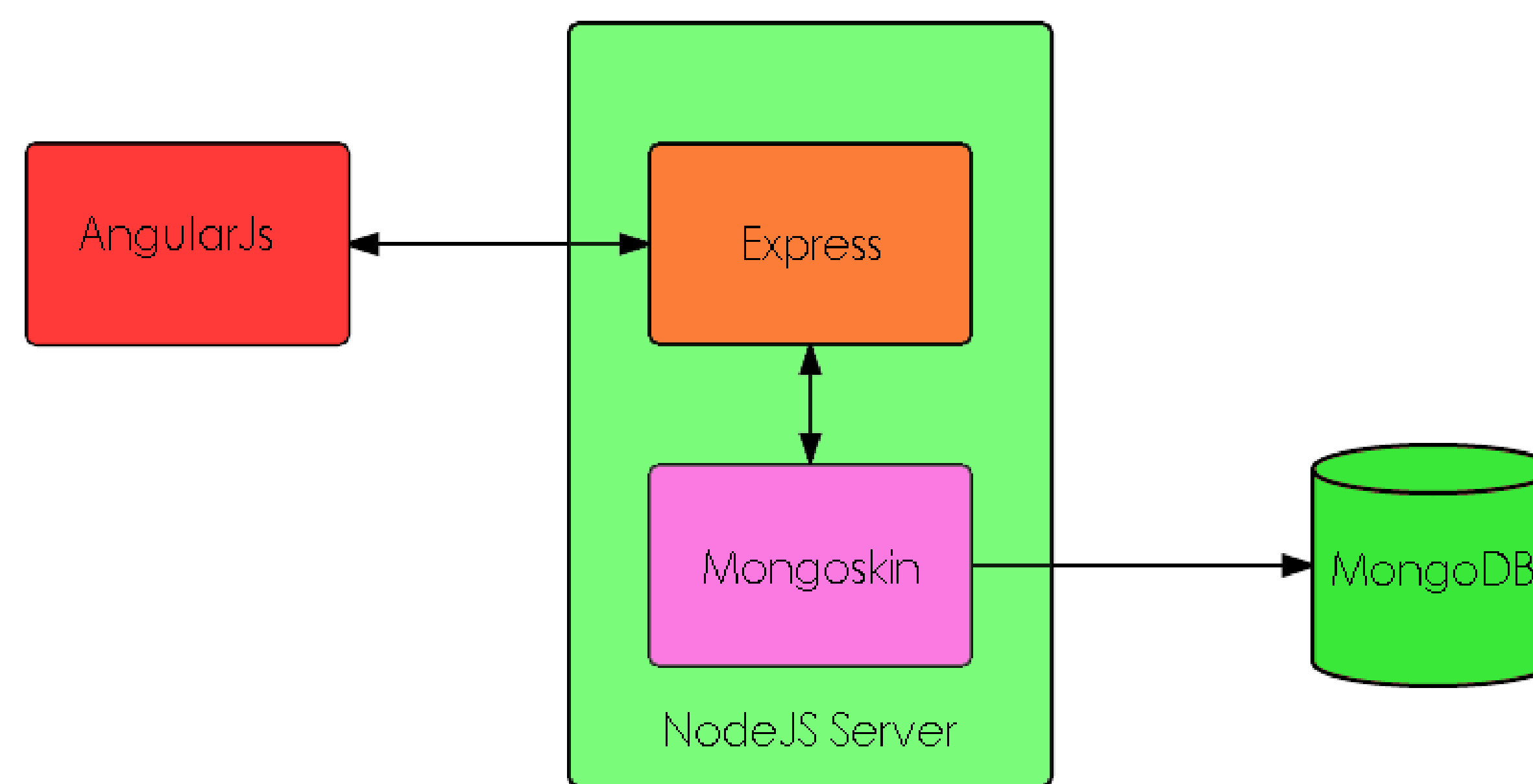
Summary

- Users can **create/edit their profile** and add contacts such as their caregivers or healthcare provider.
- Patients/caregivers can **view resources** while health care providers can view and add resources for patients.
- Users can **post to the forums**, comment on posts, add events, and opt to join an event.
- Admins can delete content.
- Users can **schedule appointments** or medications.

Screenshots



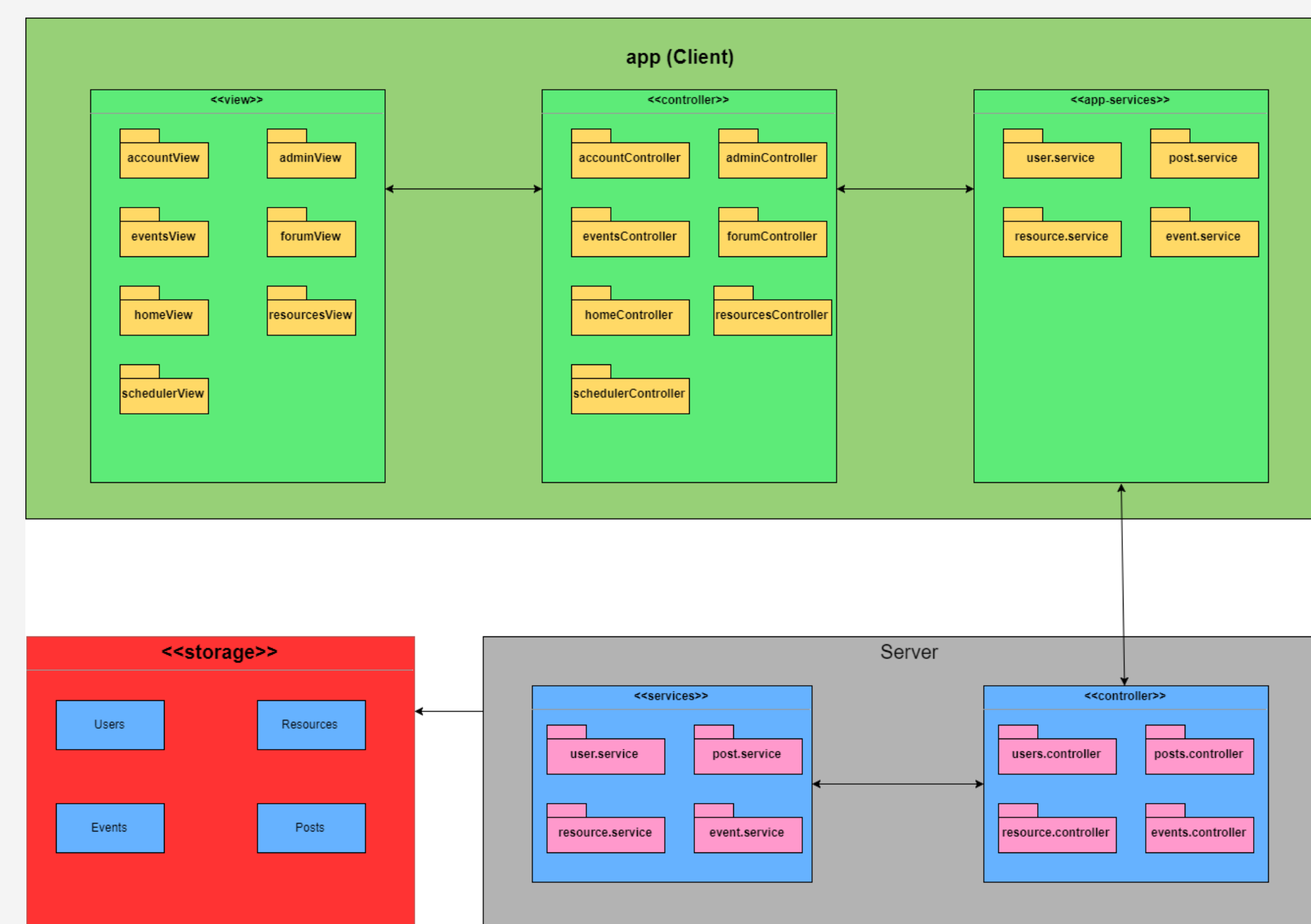
Implementation



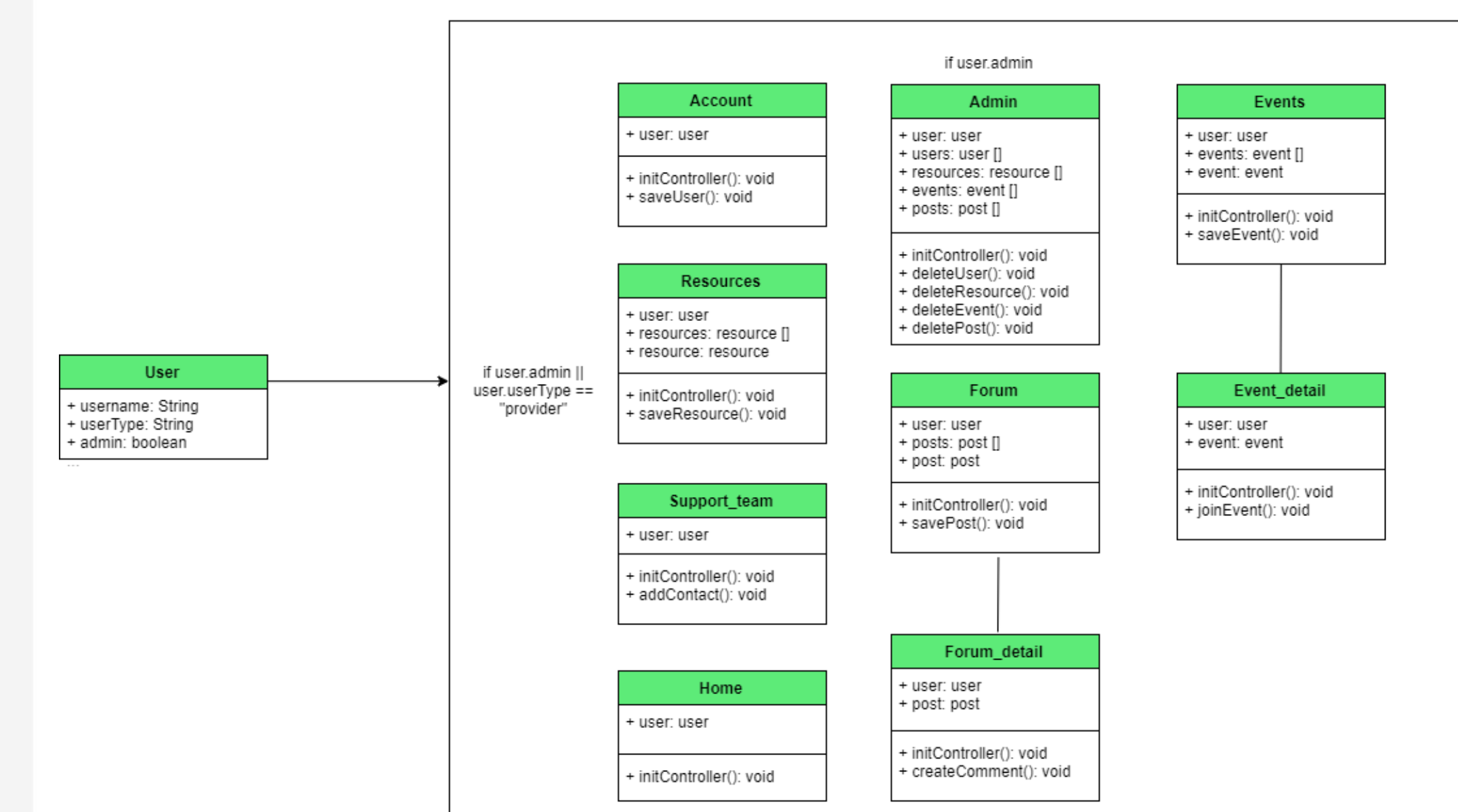
Implemented with MEAN Stack

- **Database:** MongoDB was used as the database.
- **Web Framework:** ExpressJS was utilized with Mongoskin to communicate with the database.
- **Web Server:** NodeJS is the platform used for the server.
- **Front-End:** Taken care with AngularJS framework.

Class & System Design



Structured similar to the MVC pattern, there are controllers and views but instead of a models layer, there is a services layer for data access and business logic. Mongoskin is used as the mongodb driver in the services layer providing a simpler interface for performing CRUD operations.



Minimal class diagram depicting the interactions between the user and each feature. Each feature contains an HTML file and controller that makes an object from the database available to the view and provides methods to support CRUD operations.