uCal: Incremental and Regression Testing

Matt Kramer, Eli Scherrer, Maneesh Tewani, Ben Collier, Drew Bostock, Hector Trevino Amaro

Classification of Components

Module description: Frontend

AppModule
Container for other components to make sure parts of the application are separated. Crucial part of the application that bootstraps other components.
Inputs: AppComponent, NavBarComponent, CalendarComponent, LoginComponent, GroupComponent, LoginSuccessComponent, SettingsComponent,TokenHandlerService, AuthGuardService, AuthService, AppRoutingModule,
Output:

AppRoutingModule
Encompasses logic for client-side routing of the angular application.
Inputs:RouterModule.forRoot(routes)
Output: RouterModule

AuthService
Checks if the user is authenticated by checking JWT and if it has expired.
Inputs: JwtHelperService
Output: isAuthenticated boolean

GroupComponent
Displays group information.
Inputs: None
Output: None

LoginComponent
Handles validation and logging in of users through email/password and Google OAuth Login.
Inputs: TokenHandler
Output:

LoginSuccessComponent
Intercepts the token credentials and stores them in localStorage.
Inputs: TokenHandlerService
Output:

NavBarComponent

Displays navbar and handles navigating to different client-side routes.

Inputs: tokenHandler

Output: clearing token for tokenHandler

SettingsComponent

Manages settings of user profile and group information.

Inputs: none

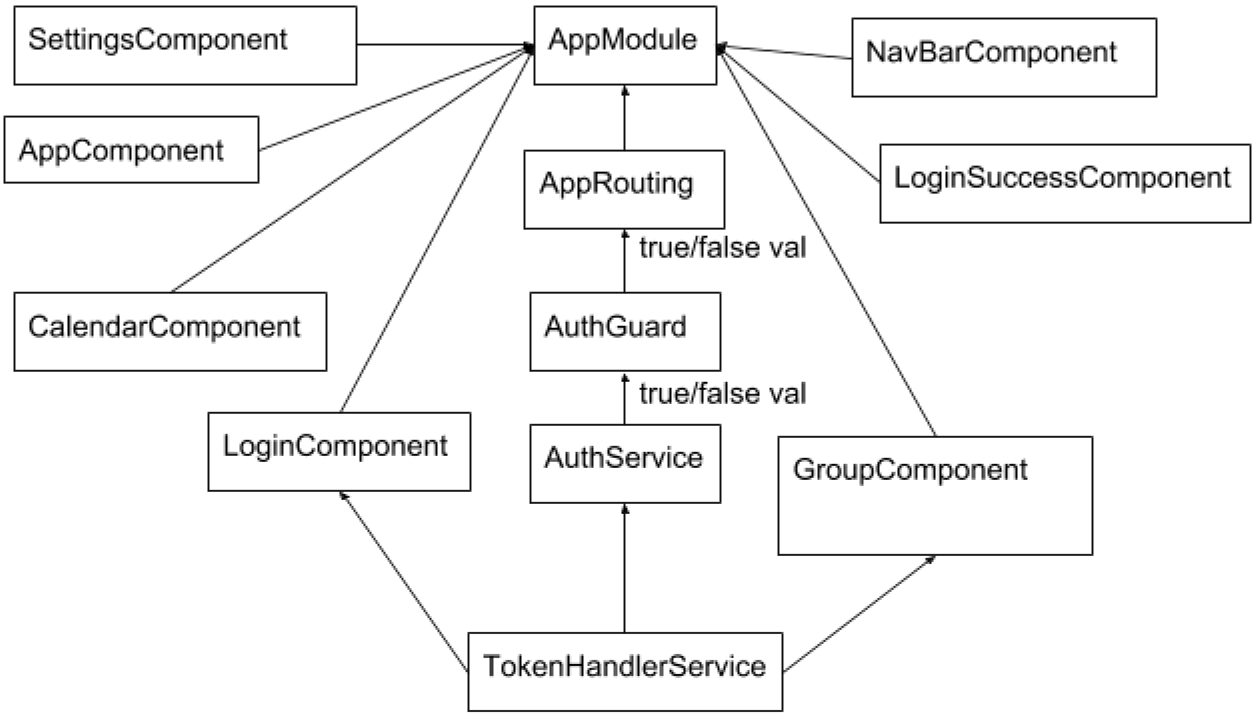
Output: none

TokenHandlerService

Sets and clears tokens for JWT authentication and interception of HTTP Requests.

Inputs: Token

Output: Local storage setting and removing tokens



Module description: Backend

Users Manager
Module in charge of adding, deleting, and modifying users and their information in the database.

Inputs: user information like email, password, and/or id
Output: confirmation message or the user object

Friends Manager
Module in charge of adding, deleting, and accepting new friends and updating the database accordingly.
Inputs: user id, friend id
Output: confirmation message

Groups Manager
Module in charge of creating, modifying, and deleting groups in the database. Also in charge of adding / removing users from the groups.

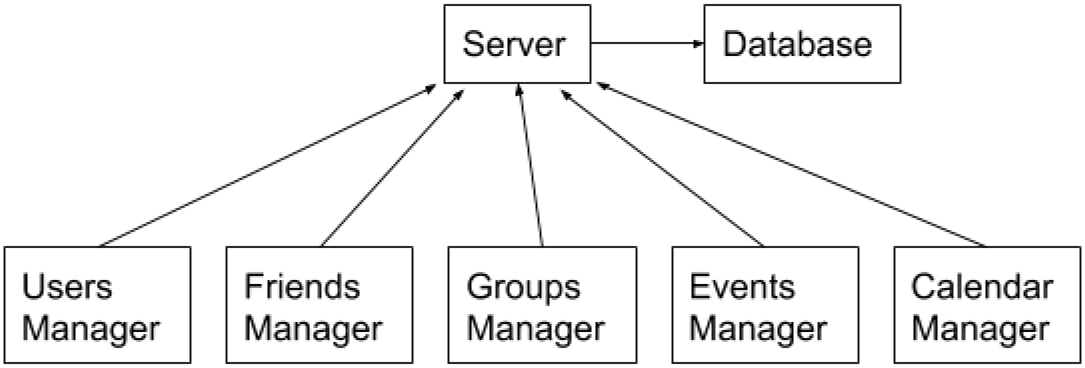
Inputs: group id, user id
Output: group information or confirmation message

Calendars Manager
Module in charge of creating, modifying, and deleting calendars in the database.

Inputs: user Id, group id, calendar id
Output: calendar object or confirmation message

Events Manager
Module in charge of creating, modifying, deleting events in the database.

Inputs: relevant event information
Output: event object



The front-end and back-end modules are connected through the endpoints which are called by the front-end.

Incremental testing technique

We used the Bottom-up testing technique. When creating our project we created each individual components that we tested on their own, then we began working on integrating them all together, and testing them together. We used drivers to simulate the functionality outside of the manager we were currently testing until the other managers were completed.

Incremental Testing

Module	**Calendar Manager**		
Defect No.	**Description**	**Severity**	**How to Correct**
1	Does not work with verifyToken	1	Add verifyToken to all routes and change the routes to grab the current user from the token instead of expecting it to be passed in the body

Module	**User Manager**		
Defect No.	**Description**	**Severity**	**How to Correct**
1	/login was returning user information	2	Returns a token instead that can be decoded for user information
2	Does not work with verifyToken	1	Add verifyToken to all routes and change the routes to grab the

			current user from the token instead of expecting it to be passed in the body

Module	**EventManager**		
Defect No.	**Description**	**Severity**	**How to Correct**
1	Does not work with verifyToken	1	Add verifyToken to all routes and change the routes to grab the current user from the token instead of expecting it to be passed in the body

Module	**GroupManager**		
Defect No.	**Description**	**Severity**	**How to Correct**
1	Does not work with verifyToken	1	Add verifyToken to all routes and change the routes to grab the current user from the token instead of expecting it to be passed in the body
2	Does not send list of invited groups	1	Implemented new invites endpoint to send a list of invites for the current user

Module	**FriendManager**		
Defect No.	**Description**	**Severity**	**How to Correct**

1	Does not work with verifyToken	1	Add verifyToken to all routes and change the routes to grab the current user from the token instead of expecting it to be passed in the body

Module	**CalendarComponent**		
Defect No.	**Description**	**Severity**	**How to Correct**

Regression Testing

Module	Defect No.	Description	Severity	How to Correct
EventManager	1	Getting an event would return 200 even if the event didn't exist, but with an empty body	2	Change the route to check and see if the event is null before returning
EventMangager	2	Updating an event's calendar does not check to make sure that the calendar id is valid	2	Add a conditional that checks to make sure that the calendar id is valid before updating the database
UserManager	3	Login returned user data instead of {message: "Success!"}	2	Replaced tests to check for {message: "Success!"} instead of user information
EventManager	4	Deleting an event would cause it to be deleted from everyone's calendars	2	Change it so that deleting an event only removes the event from the specifies calendar so that other people who still want the event still have access to it

UserManager	5	/login responded with 500 instead of 401 Unauthorized	2	Changed status code in users.js and server.js as well as in tests
EventManager	6	Events could be added multiple times to one calendar	2	Added a check to see if the event id was already in the list of events for the calendar you are attempting to add to
EventManager	7	Users could be in the accepted and declined rsvp arrays at the same time if they accepted then declined	2	Remove the user from the opposite array when accepting/ declining an rsvp
EventManager	8	Event routes did not work with verify token (found during incremental testing)	2	Add verifyToken to all routes and change the routes to grab the current user from the token instead of expecting it to be passed in the body