



Student: Andres Moser, Florida International University
Mentor: Jose Ponce, Ultimate Software
Instructor: Masoud Sadjadi, Florida International University

Problem

- Semesters were not manageable
- Lack of semester effect for Admins and Faculty
- Unable to announce new semesters

Solution

- Semesters have a new Maintenance tab in Admin Tools
- Filters based on semester properties
- Email Service to Previous Product Owners

Current System

- Only able to set Semester as 'Active' or Disabled
- No semester properties outside of 'Active' and 'Disabled'
- No way for Admin to alert Product Owners

Figure 1: Semester Maintenance

Semester	Start Date	End Date	Current Semester	Viewable	Proposable	Applicable
Fall 2016	Sep 1, 2016	Jan 1, 2018	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Spring 2017	Feb 1, 2017	Jun 1, 2017	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Summer 2017	Jun 1, 2017	Sep 1, 2017	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fall 2017	Sep 1, 2017	Jan 1, 2018	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Figure 2: Email Product Owners about Semester

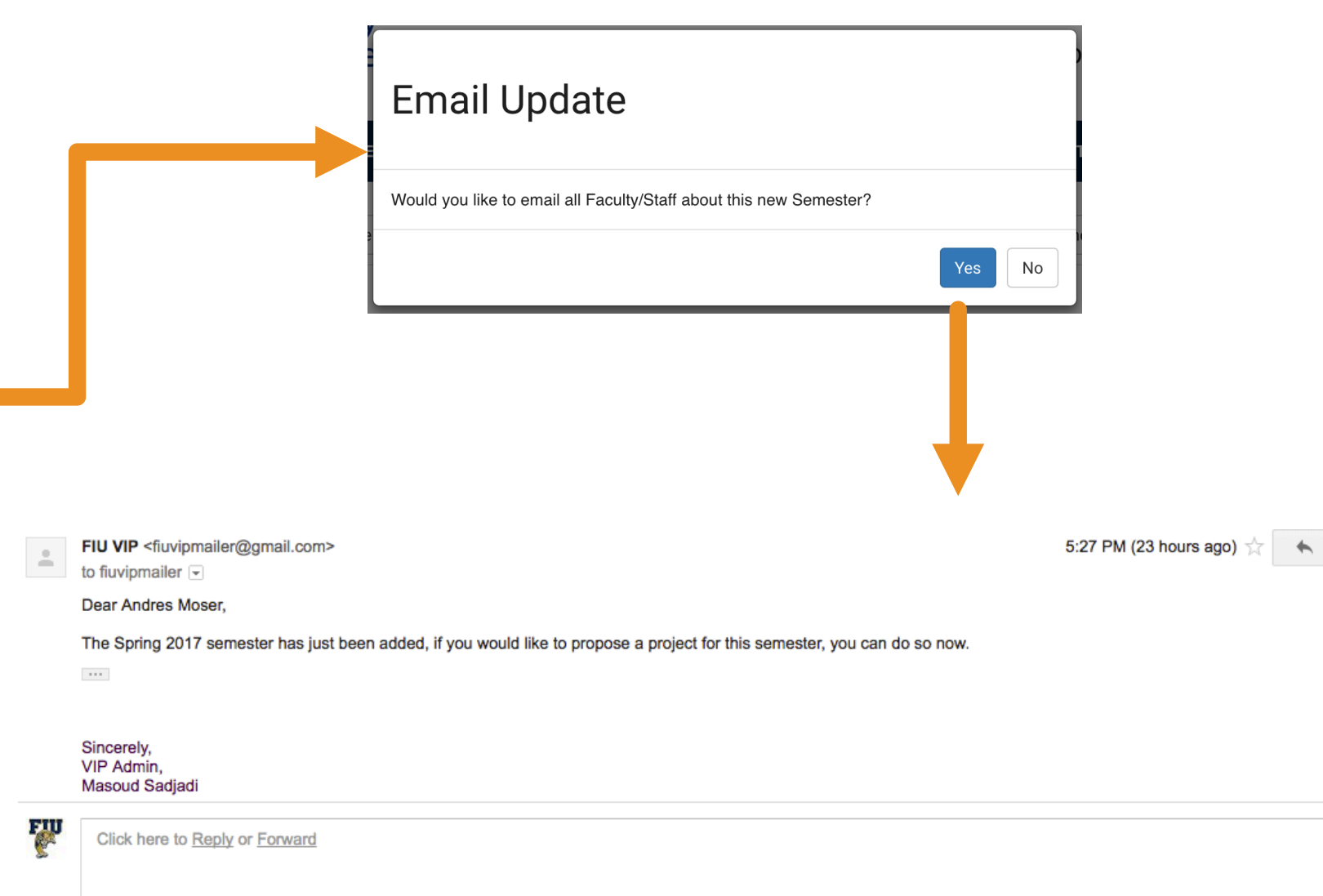


Figure 3: Add and Edit Semester

Add New Semester

Semester Name:

Start Date: End Date:

Current Semester: On Off

Viewable by Students: On Off

Proposable by Faculty: On Off

Applicable by Students: On Off

Edit Semester

Semester Name:

Start Date: End Date:

Current Semester: On Off

Viewable by Students: On Off

Proposable by Faculty: On Off

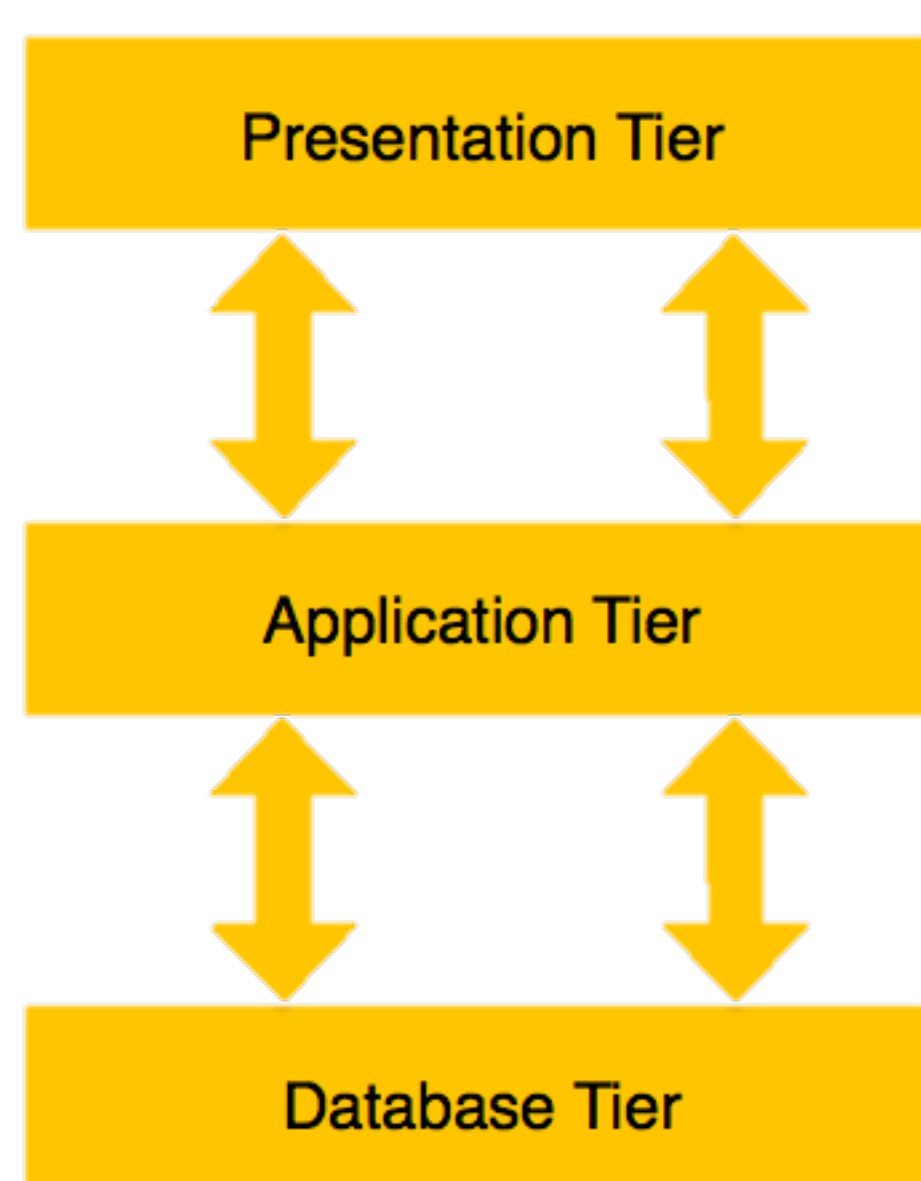
Applicable by Students: On Off

Requirements

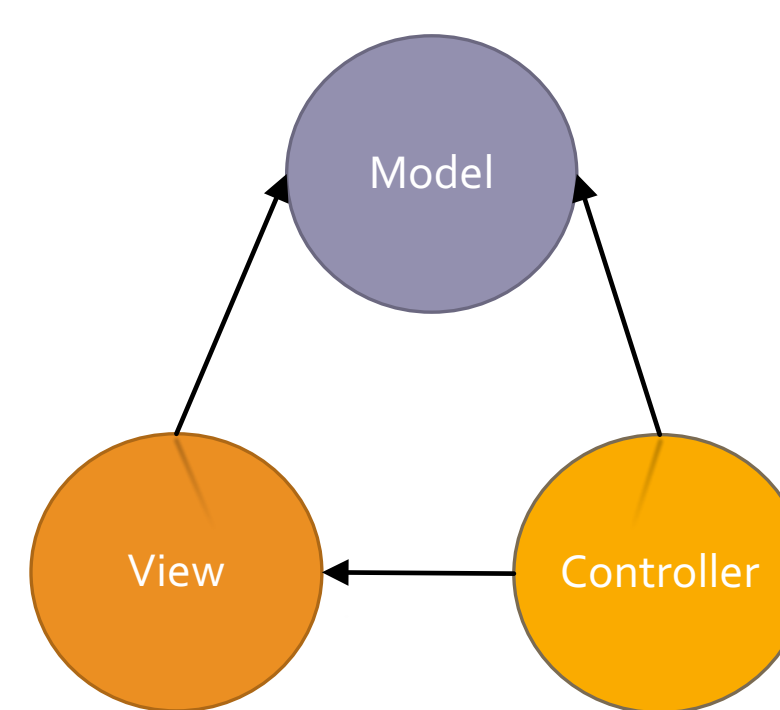
- Admin ability to Create/Manage/Delete Semesters
- Consistent UI through Admin Maintenance tabs
- Default Semester for Admins
- Filters based on Semester statuses
- Email Product Owners about new semesters

System Design

Three-Tiered Architecture



Model-View-Controller



Implementation

M **mongoDB**
NoSQL Database

E **express**
App Framework

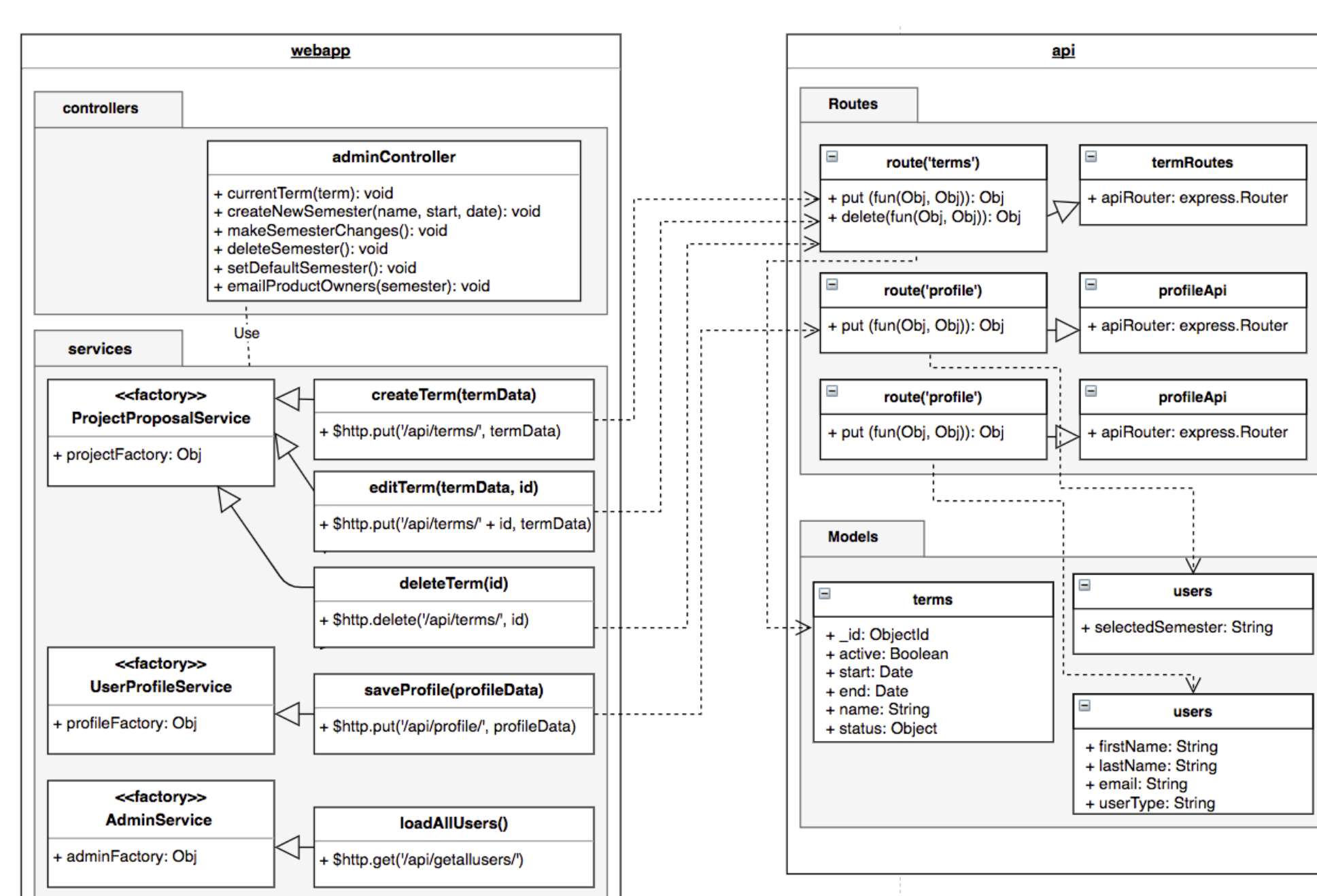
A **ANGULARJS**
JavaScript Framework

N **node**
JavaScript Runtime

Verification

- Testing done locally for each User Story with a mix of Unit and Integration Tests
- User Stories merged onto a develop server as they are completed so the entire system can be tested

Object Design



Summary

- Semester was redefined and made manageable by Admins
- Admin Panel has a consistent improved design
- Admin can now notify product owners about semester information

Acknowledgement

The material presented in this poster is based upon the work supported by Dr. Masoud Sadjadi and Dr. Francisco Ortega. I am thankful to the help that I received from my group members Leon Liang, Dale Keith, Steve Hirabayashi, and Adam Levy.