

MICHIGAN STATE

U N I V E R S I T Y

Project Plan

Group Project Organization and Scheduling

The Capstone Experience

Team MSU ITS

Sarah Abumansoor

Jacob Bickel

Kristin Calder

Jacob Dasuqi

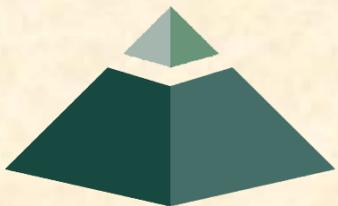
Cyndy Ishida

Jack Wydra

Department of Computer Science and Engineering

Michigan State University

Spring 2019



*From Students...
...to Professionals*

Functional Specifications

- All students work on group projects at Michigan State
 - Frustration around communicating with members, scheduling times to meet, keeping track of used resources, etc.
- Web application designed to centralize scheduling and any services used to work on the project
 - Create and maintain multiple groups
- Scheduling feature finds the best time for all group members to meet
 - Pair your calendar applications (Google Calendar or Office 365)
 - Scheduler grabs all blank times from all members' calendars and suggest multiple times which most/many members can meet
 - Users select their preferred meeting time, which is added to the group calendar
- Quick access to collaboration tools
 - Instantly create connections to services that handle messaging, documents, etc.
 - Suggestions about possible services the group may find useful for their project

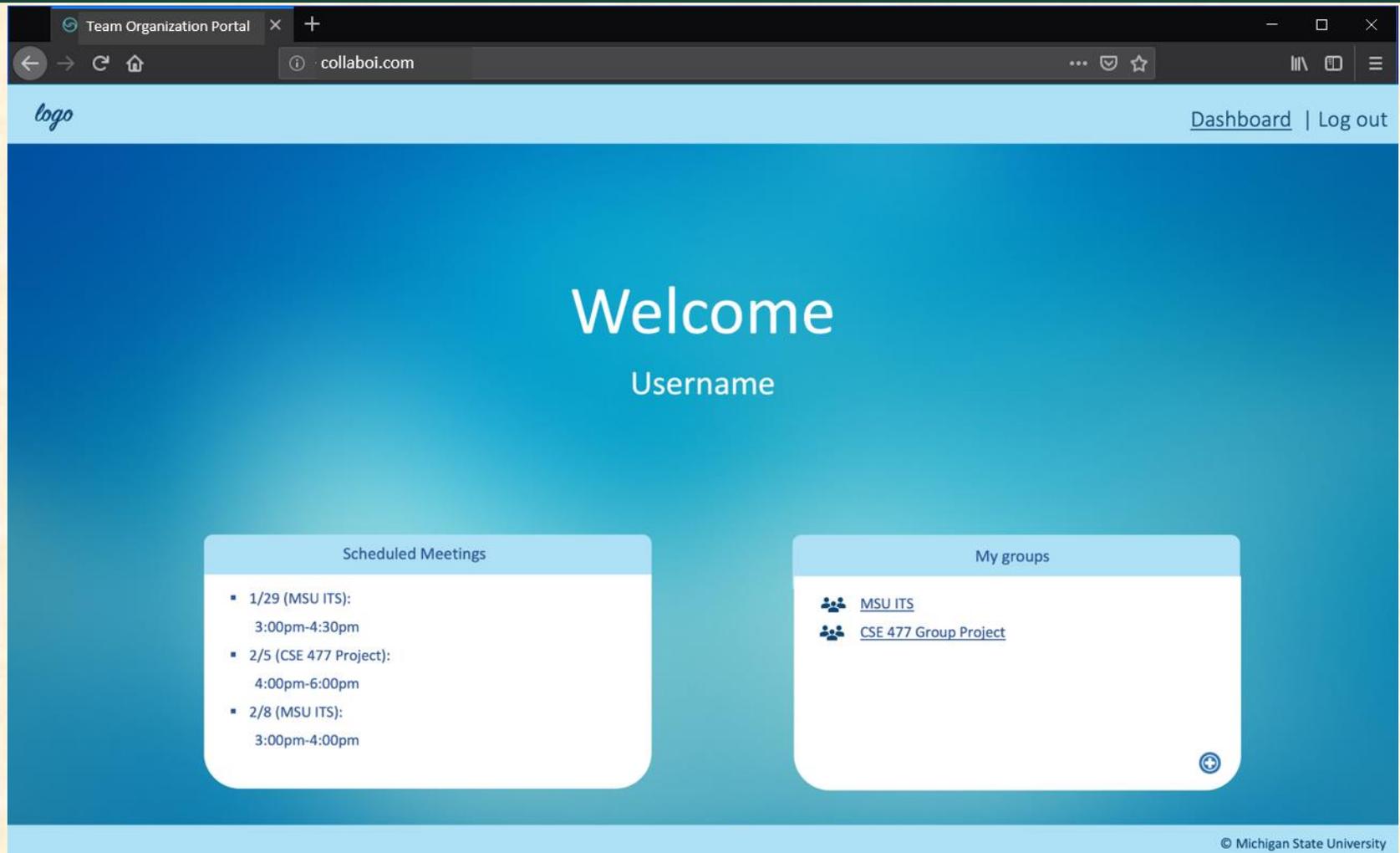


Design Specifications

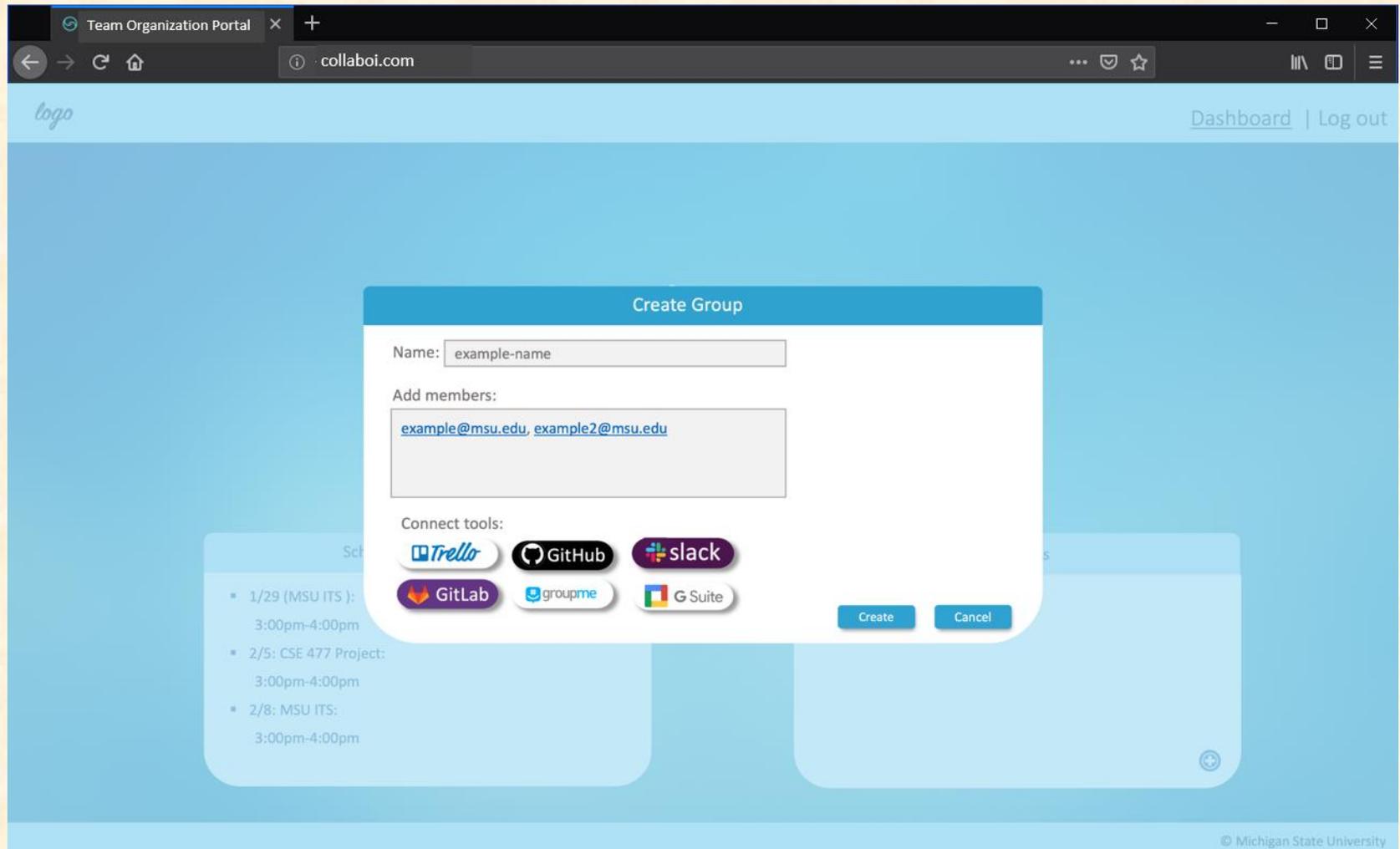
- Website elements:
 - User Portal
 - Managing Groups: create/view groups
 - List of all scheduled meetings
 - Team Portal
 - Meeting Scheduler: time table to schedule new meetings
 - Collaboration Tools
- UI Components
 - Perceivable: UI components simple and perceivable
 - Operable: help users navigate content easily
 - Understandable: media and text content easy to understand



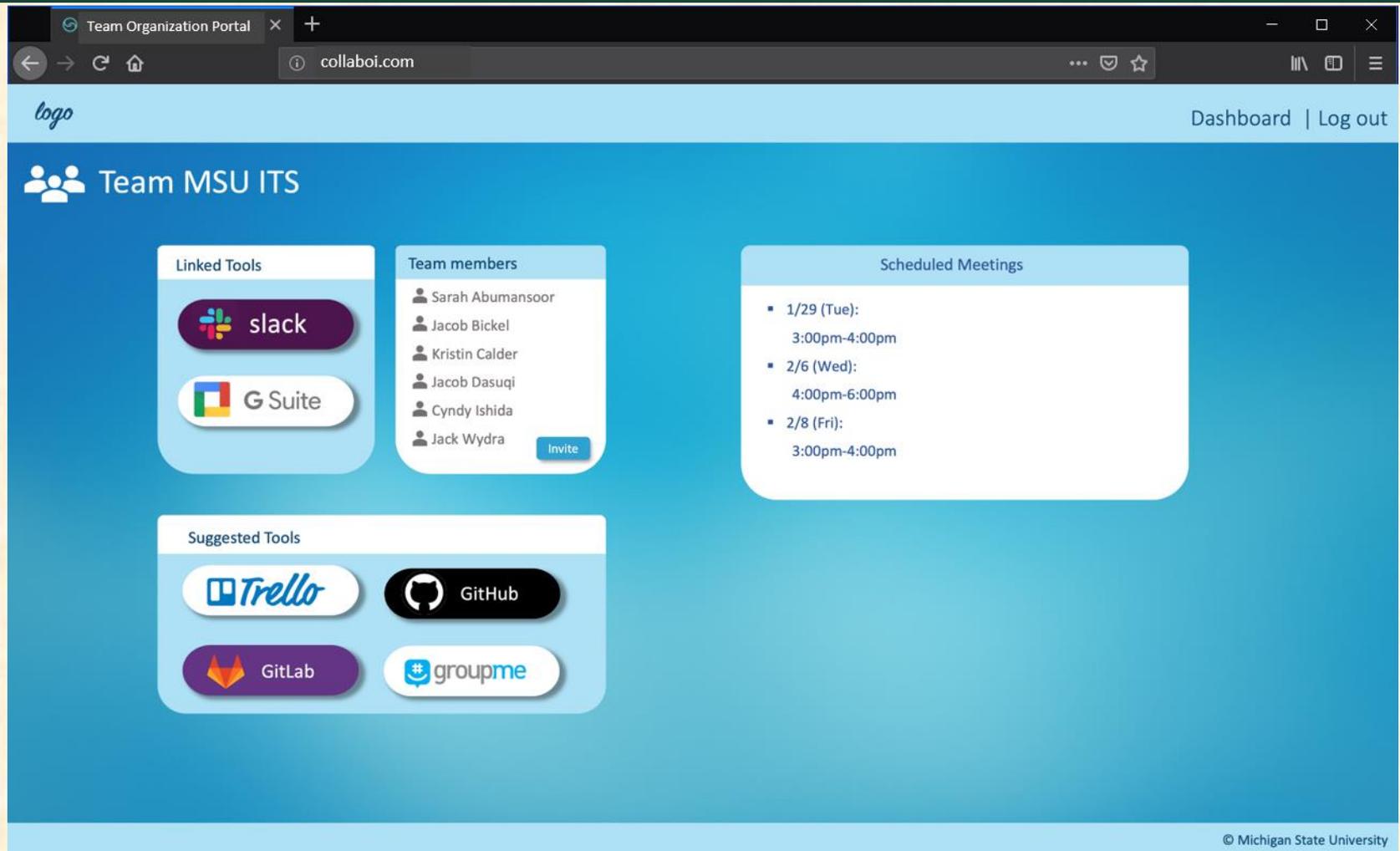
Screen Mockup: Personal Portal



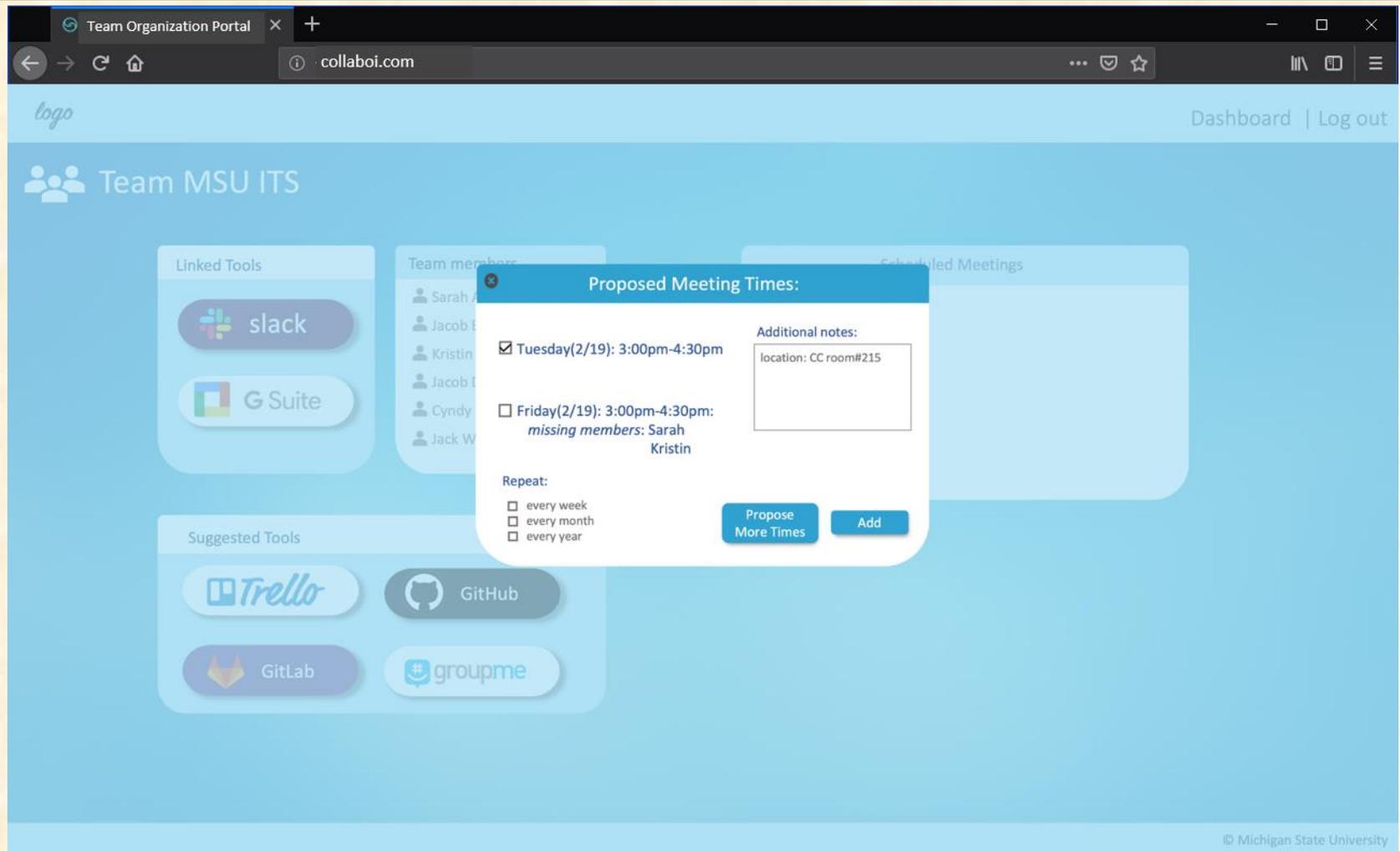
Screen Mockup: Group Creation



Screen Mockup: Team Portal



Screen Mockup: Team Scheduler

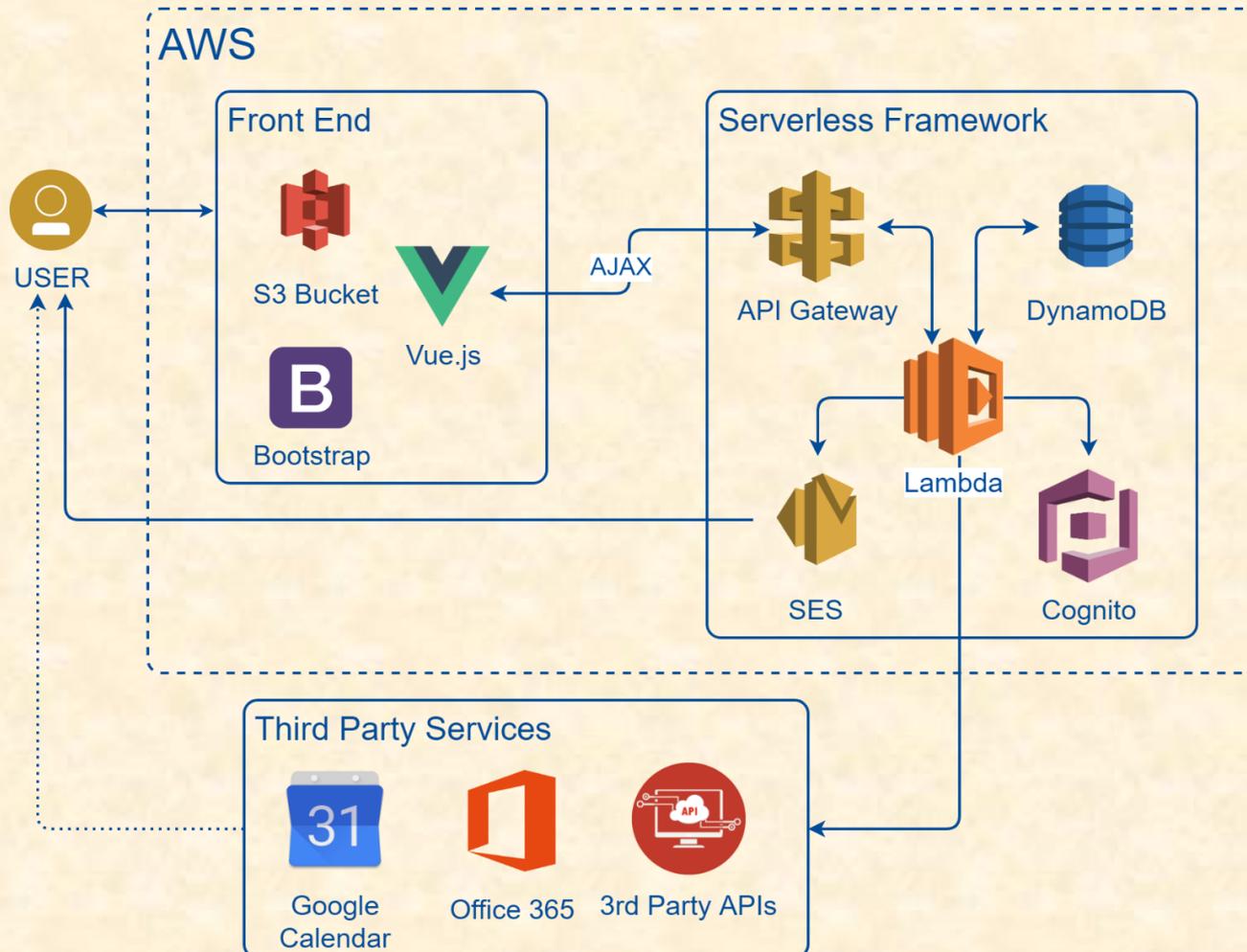


Technical Specifications

- Serverless Framework
 - Python 3.7 Runtime
 - Amazon API Gateway
 - Lambda
 - DynamoDB
 - Amazon SES
 - Cognito User Control
- Vue.js / Bootstrap
- Integration with 3rd Party APIs via OAuth



System Architecture



System Components

- Software Platforms / Technologies
 - Front-end: Bootstrap and Vue.js
 - Back-end: AWS Serverless Infrastructure
 - Lambda, DynamoDB, Cognito user control, SES
 - API Gateway and OAuth – Integration with numerous 3rd Party APIs
 - GitLab – Version Control and hosting code for web application



Risks

- Integrating with 3rd Party APIs
 - Utilizing many 3rd party API services and must ensure consistent usability
 - Great DevOps, maintain validation by automating local tests and generalizing implementation to support future updates to the APIs
- Security Concerns Storing User Information
 - System will store as little user information as possible to avoid security liability
 - Only store login information, delegate other info to authentication APIs
- Support Multiple Calendars
 - Ability to interpret multiple calendars for availability i.e., Google Personal Calendar and Outlook Work Calendar
 - Identify the most common use case and stick to a well defined and well documented approach
- Consistent User Experience for Non-MSU Students
 - Avoid confining users to MSU students
 - Design from the beginning for the general user with a .edu email



Questions?

?

?

?

?

?

?

?

?

?

