

# Communication Tool Iteration #2

Group 1, Team 2:

Oscar Ingham, William Wilson, Shishuang Shu, Yuyan Zhang, Xinzhu Xu

# Roles

Bill: Team Leader

Oscar: Implementation Leader

Jenna and Sherry: Front end & Functional Tests

Bill and Jewel: Backend Team

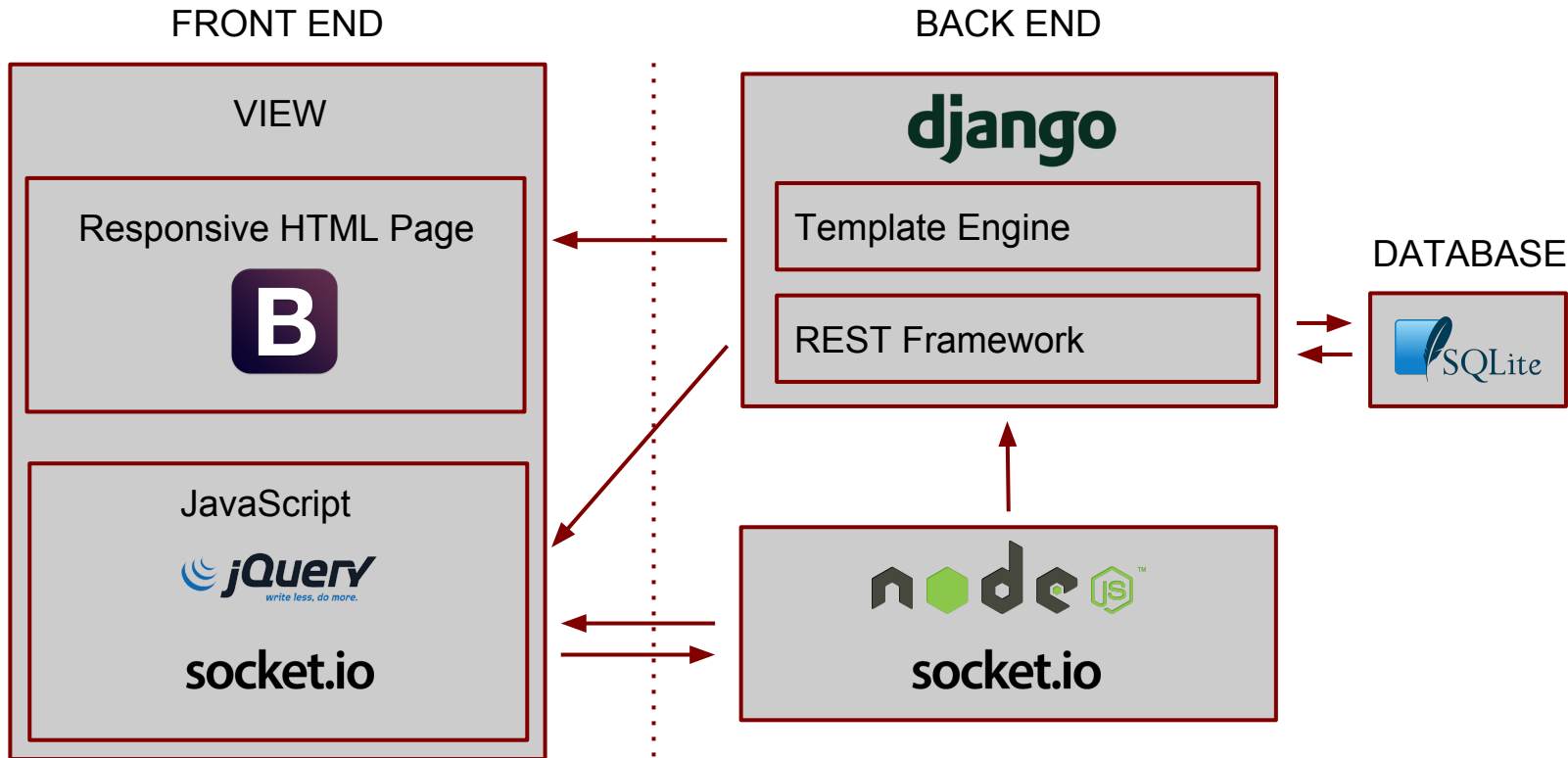
# Project Requirements for Iteration #2

- Add multi-room messaging
- Show number of new messages
- Responsive UI especially for mobile devices
- Archive conversations in the database.
- Change development environment to match production. Refactored front end code.

# Application Architecture

- Single HTML page served through Django
- Data managed using AJAX and Django Rest Framework
- SocketIO for real-time message sending
  - Separate NodeJS service
  - Manage collections of websockets by namespace
- Persistence using SQLite (for now)

# Architecture



# Environment

- Moved from Ubuntu to CentOS 6.6
- Gunicorn is our WSGI HTTP server
- NGINX acts as our reverse proxy and static file server
- Supervisor keeps both node and gunicorn running

# Implementation - Back End

- Used SocketIO namespaces to create user groups.
  - Node server initially queries database for namespace names and opens a socket for each namespace
- Django RESTful-API endpoint established to get room data based on the user

# Implementation - Front End

1. **Responsive: design different layout for different sizes of screen(Full site, mobile site, tablet site).**
2. **The headers and conversation content will change with the the rooms that user choose.**
3. **User can create new rooms (in progress)**



## Chatting Room

+ Create New Teams

General

Team One

Team Two

Team Three

Team Four

## Users

Oscar

Bill

Sherry

Jenna

Jewel

## Quick link to

Requirement Management Tool

Issue Management Tool

# General

Options ▾

Search messages...



User 76: user 1  
User 76: user 1  
User 23: user 2  
User 23: user 2  
User 76: user 1  
User 76: user 1  
User 23: user 2  
User 23: user 2  
User 50: sendcleartext button test  
User 50: sendcleartext button test  
User 50: return key press test  
User 50: return key press test  
User 50: general room  
User 50: general room  
User 83: cp size test  
User 83: cp size test  
User 83: mobile size test  
User 83: mobile size test


Enter a message...



Communication tool


52.4.50.51

Apps Shopping Yuyan Other Bookmarks



# General

Menu Options

Search messages... 

User 76: user 1

User 76: user 1

User 23: user 2

User 23: user 2

User 76: user 1

User 76: user 1

User 23: user 2

User 23: user 2

User 50: sendcleartext button test

User 50: sendcleartext button test

User 50: return key press test

User 50: return key press test

User 50: general room

User 50: general room


User 83: cp size test


User 83: cp size test

User 83: mobile size test

User 83: mobile size test

Enter a message...







Communication tool


52.4.50.51


Apps Shopping Yuyan Other Bookmarks


## Chatting Room


 Create New Teams

 **General**


 Team One


 Team Two


 Team Three


 Team Four


## Users

 Oscar

 Bill

 Sherry

 Jenna

 Jewel

## Quick link to

 Requirement Management Tool

 Issue Management Tool

# Testing Cases(Functional Testing)

Test case 1: test mobile site and PC site

Test case 2: test all buttons

Test case 3: test general message sending function

Test case 4: test group message sending function

# Testing Cases,cont.

video demo

# Challenges

- Moving operating systems
- Maintaining consistent configuration for individual VMs
- Trying to create endpoints to acquire serializing data
- Interact with user management with Team1

# Next Iteration

- User model Integration
- Private text messages
- Record Searching
- Emoticons
- @mentions
- Unit tests

# References

Bootstrap: <http://getbootstrap.com/>

SocketIO: <http://socket.io/>

NodeJS: <http://nodejs.org/>

Django: <http://www.djangoproject.com/>

DRF: <http://www.django-rest-framework.org/>

**LIVE DEMO: <http://52.4.50.51/>**

