Software Design Description for Mashbot

George D'Andrea Andrew Gall Josiah Kiehl Cody Ray Vito Salerno

January 30, 2010

Revision History

Name	Date	Reason for Changes	Version
George D'Andrea, Andrew Gall, Josiah Kiehl, Cody Ray, Vito Salerno	17 January 2010	Initial Version	1.0

Contents

1 Introduction

1.1 Purpose

This document serves to expand upon the requirement into implementation details and technology choices. This document should be referenced when specific features are being implemented.

1.2 Scope

This covers the general architecture of Mashbot, as well as the design decisions used to apply that architecture via various appropriate technologies, libraries and frameworks.

1.3 Definitions, Acronyms, and Abbreviations

- MVC — Model View Controller

1.4 Context Diagram

2 Architecture

2.1 Overview

Generally, Mashbot will be implemented using a strict Model-View-Controller architecture. This is augmented by the inclusion of the Presentation and Aggregation Platform, the purpose of which is to abstract the interaction with external service APIs from the application as a whole, thus allowing a pure MVC architecture to be implemented, increasing maintainability, extensibility and accessibility.

2.2 Four-Tier Architecture

- Data Layer / Model
- Presentation Layer / View
- Business Layer / Controller
- Publishing and Aggregation Platform

2.3 Service-Oriented Architecture

Mashbot will be implemented as two distinct yet related services. The Campaign Manager will handle the interaction between the user and the data the Campaign Manager is concerned with, where the Presentation and Aggregation Platform will handle the interaction between external service APIs and the Campaign Manager.

2.4 Survey of Technologies Used

2.4.1 Campaign Manager

- Presentation Layer
  - HAML — HTML replacement markup language, for building web layout structure.
  - SASS — CSS replacement stylesheets, for applying visual styles to the layout built in HAML.
  - jQuery — JavaScript library which provides cross-browser compatibility as well as streamlined Ajax request handling.

  - Google Chart API — Public service provided by Google which generates many different kinds of charts and graphs.

- Business Layer
  - Ruby — Dynamic programming language.
  - Rails — Web application framework written in Ruby which provides a concise Model-View-Controller architecture.
  - Heroku — Rails engine which provides enhanced production deployment via Rails compilation, a fast readonly filesystem, and horizontal scaling.

- Data Layer
  - ActiveRecord — Component of Rails which provides the Active Record pattern of data access, creating data model objects and relationships for interacting with resources in a database.
  - MySQL — Fast and free relational database which plugs into Rails without effort.

2.4.2 Publishing and Aggregation Platform

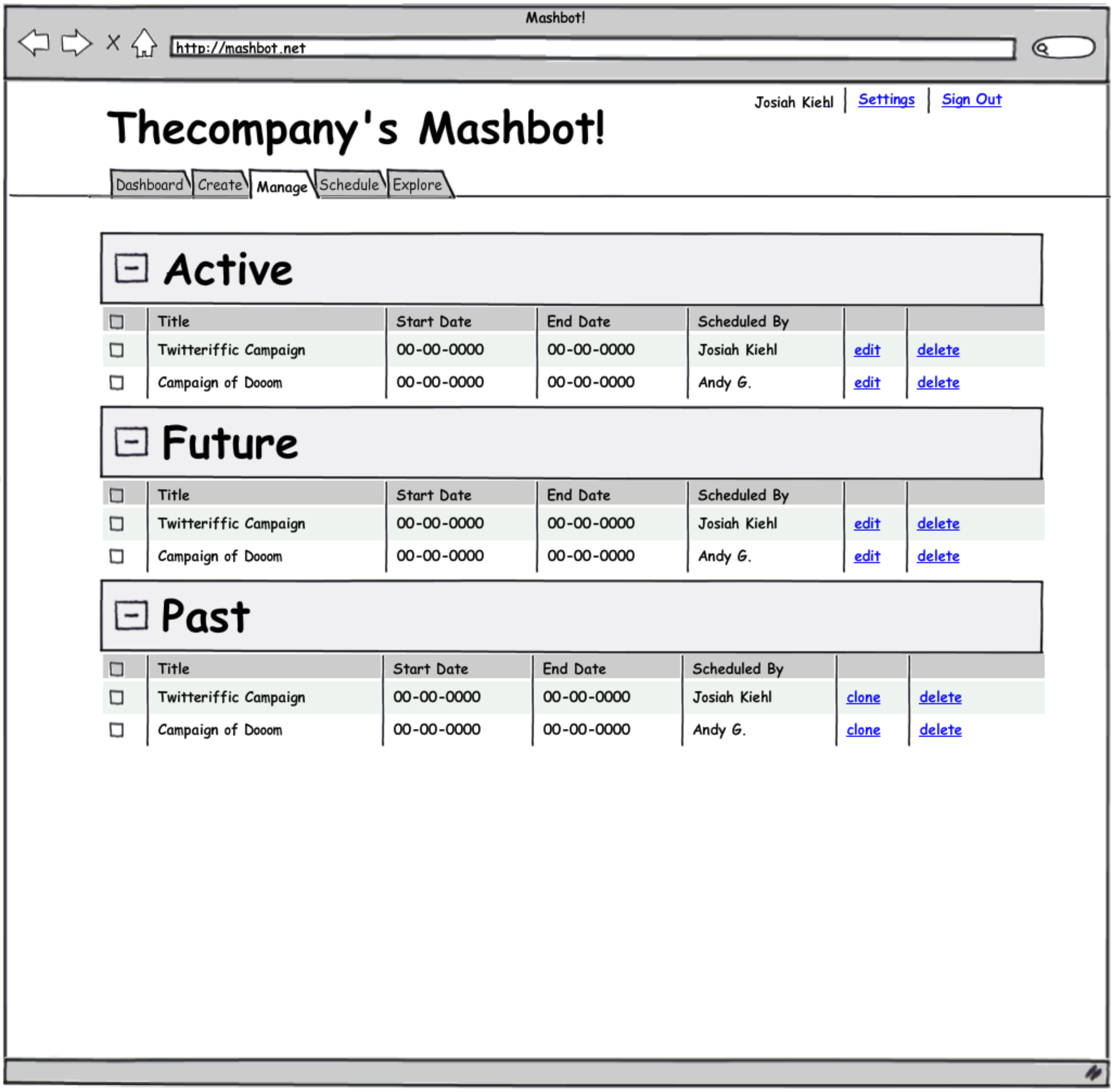
- TODO:TODO:TODO:TODO:TODO:TODO:TODO:TODO:TODO:TODO:TODO:TODO:TODO:TODO:TODO:TOD
  Put tech stuff here.

2.5 Presentation Layer Components

2.5.1 Campaign Views

Campaigns are accessed via the Create and Manage tabs on the primary navigation tabs. Create is for the Create view, Manage is for List, Show and Edit.

- Create — This is where users can create new campaigns.
- List — This is where users can view, update or delete existing campaigns.



Thecompany's Mashbot!

Active

□	Title	Start Date	End Date	Scheduled By		
□	Twitteriffic Campaign	00-00-0000	00-00-0000	Josiah Kiehl	edit	delete
□	Campaign of Dooom	00-00-0000	00-00-0000	Andy G.	edit	delete

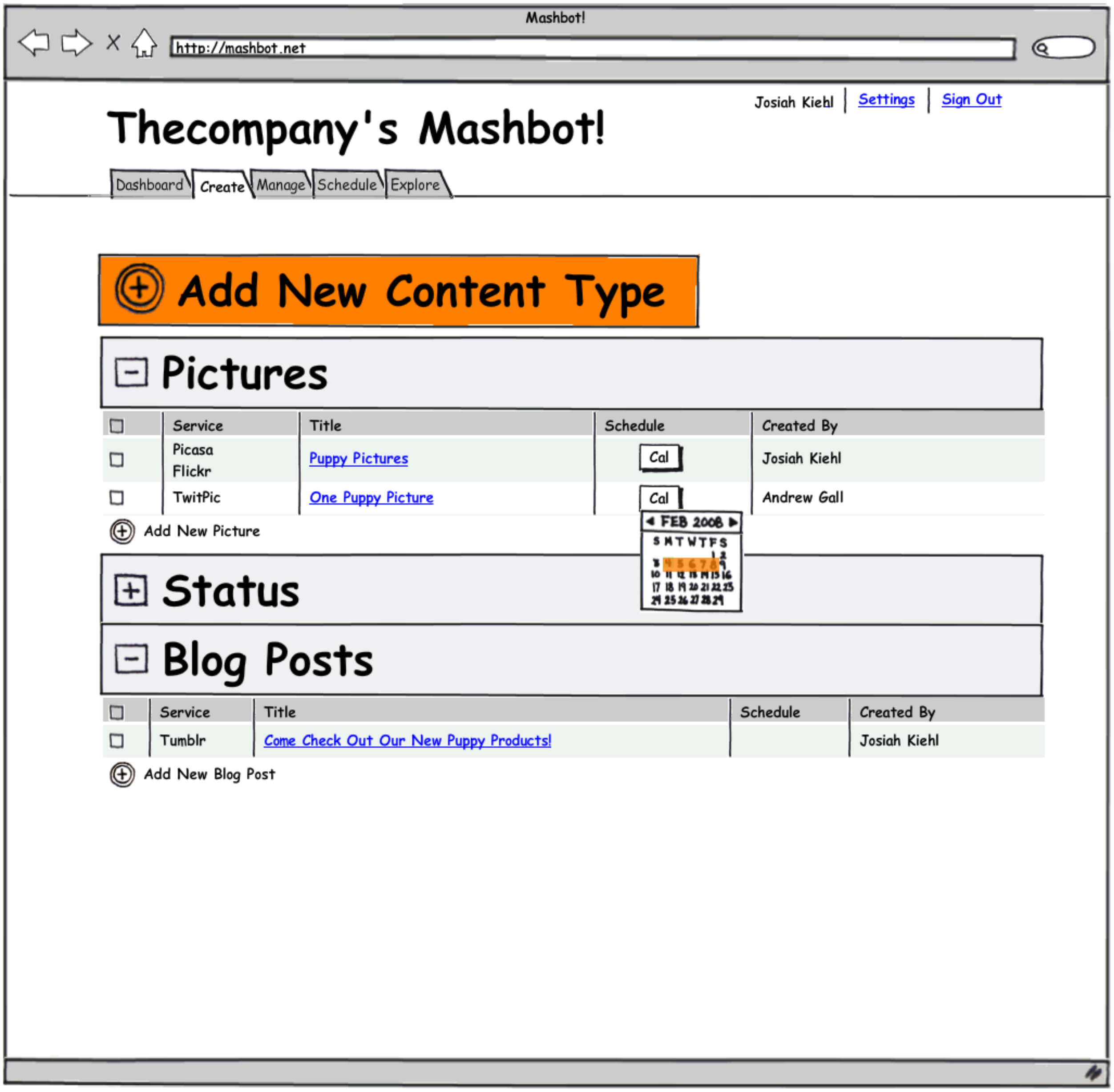
Future

□	Title	Start Date	End Date	Scheduled By		
□	Twitteriffic Campaign	00-00-0000	00-00-0000	Josiah Kiehl	edit	delete
□	Campaign of Dooom	00-00-0000	00-00-0000	Andy G.	edit	delete

Past

□	Title	Start Date	End Date	Scheduled By		
□	Twitteriffic Campaign	00-00-0000	00-00-0000	Josiah Kiehl	clone	delete
□	Campaign of Dooom	00-00-0000	00-00-0000	Andy G.	clone	delete

- Show — This view is what is shown when the user wants to view an existing campaign via the Show view. This is also where the Content pieces will be listed.



⊟ Pictures

□	Service	Title	Schedule	Created By
□	Picasa Flickr	Puppy Pictures	Cal	Josiah Kiehl
□	TwitPic	One Puppy Picture	Cal	Andrew Gall

⊕ Add New Picture

⊞ Status

⊟ Blog Posts

□	Service	Title	Schedule	Created By
□	Tumblr	Come Check Out Our New Puppy Products!		Josiah Kiehl

⊕ Add New Blog Post

- Edit — This is virtually the same view as Create, however this will be prepopulated with the existing content of the given Campaign.

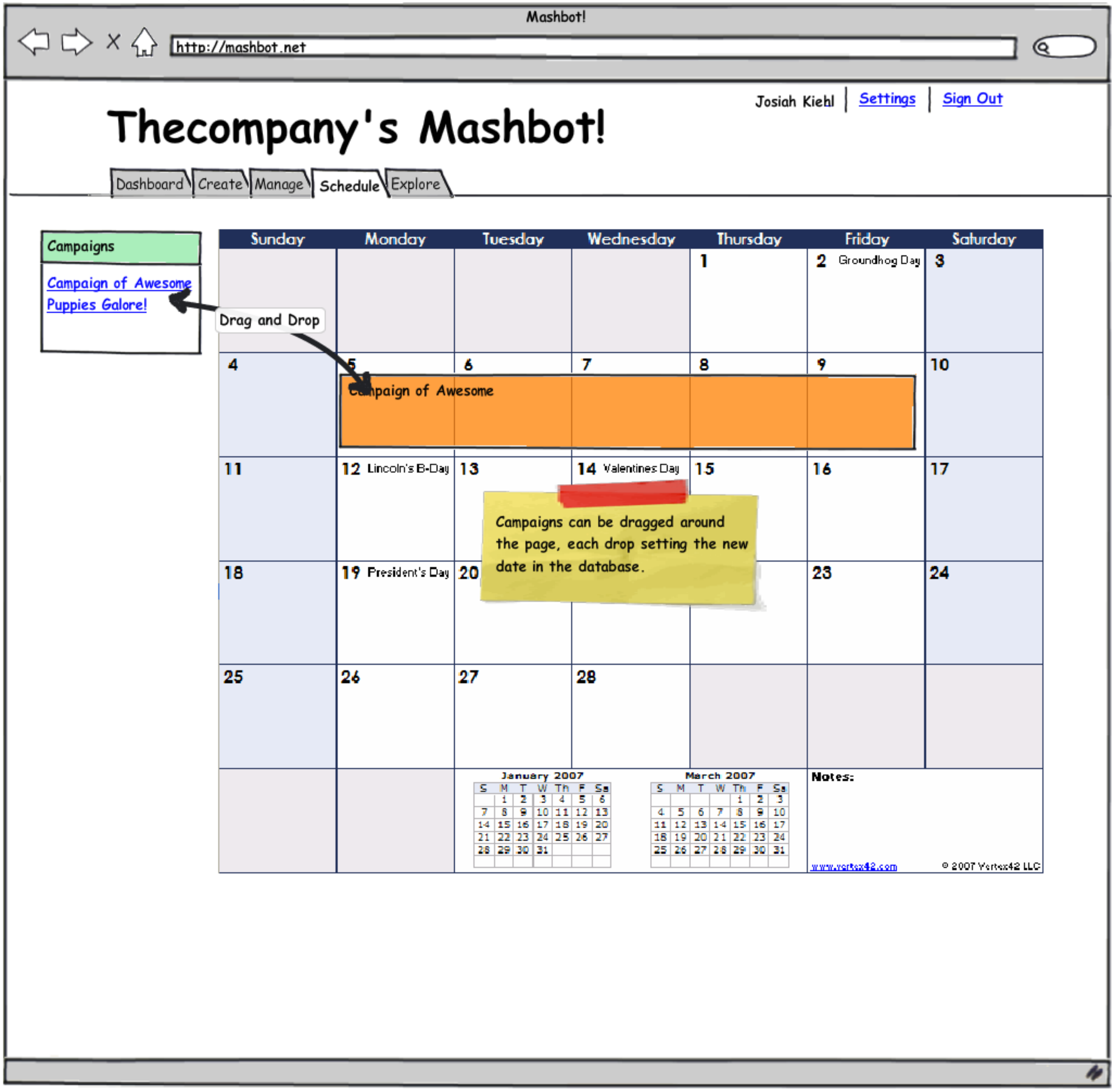
2.5.2 Content Views

Content pieces are included inside Campaigns. These views are accessible via the Show view of a Campaign for the corresponding Campaign id.

- Create — When on the Show view of a given campaign, the user can enter the Create view for Content.
- Show — This is how the user previews the Content they have created.
- Edit — This is virtually the same view as Create, however this will be prepopulated with the existing content of the given Content.

2.5.3 Scheduling Views

- Primary Scheduling View — consists of a list of Campaigns available to be scheduled (ie: they do not have existing start/stop dates) as well as already scheduled Campaigns placed properly on the calendar.



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1	2 Groundhog Day	3
4	5 Campaign of Awesome	6	7	8	9	10
11	12 Lincoln's B-Day	13	14 Valentines Day	15	16	17
18	19 President's Day	20			23	24
25	26	27	28			

Campaigns can be dragged around the page, each drop setting the new date in the database.

January 2007

S	M	T	W	Th	F	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

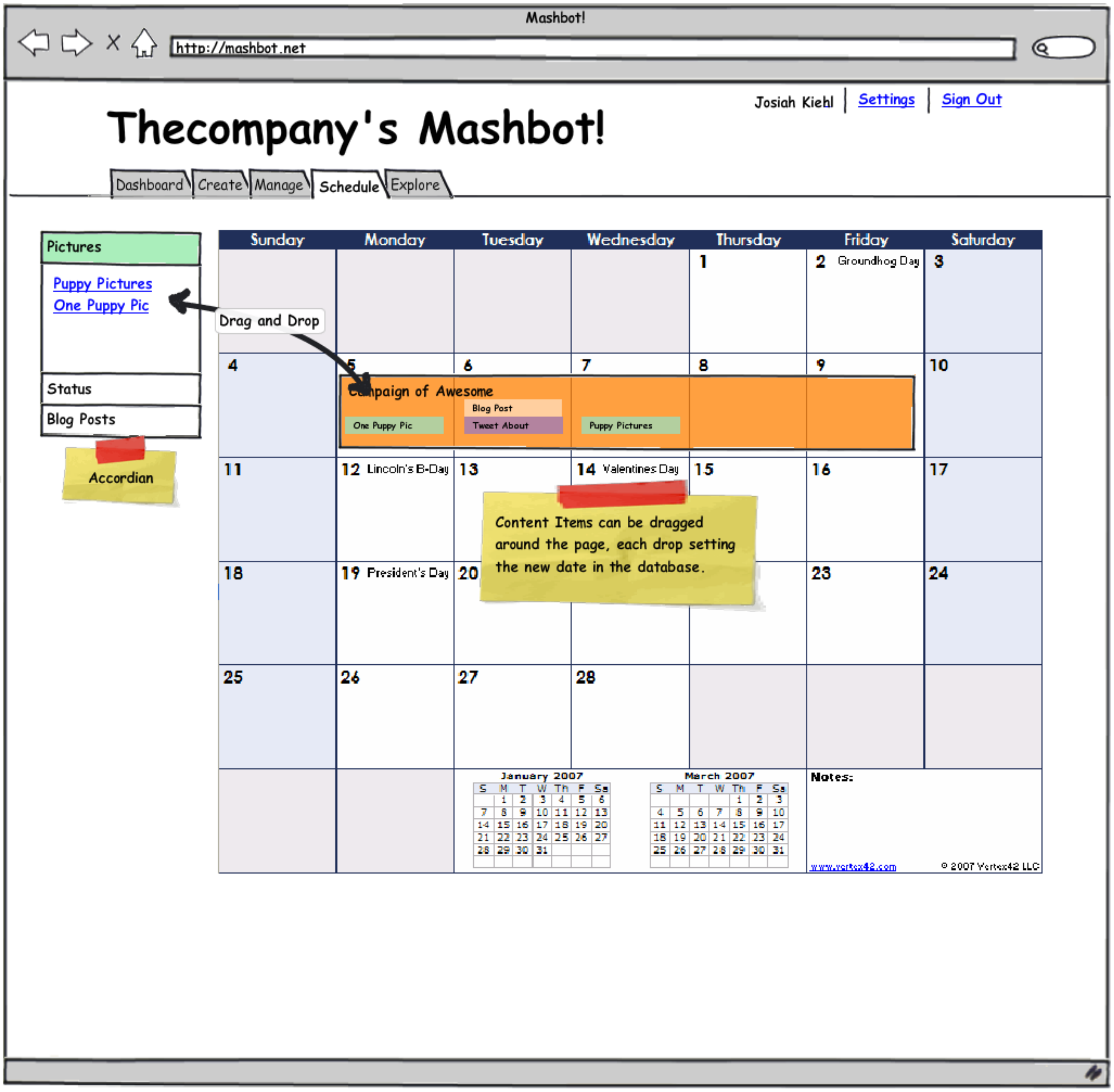
March 2007

S	M	T	W	Th	F	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Notes:

www.vertex42.com © 2007 Vertex42 LLC

- Content Scheduling View — similar to the Primary Scheduling View, however the items available to be scheduled here are the individual content pieces of the Campaign. This is accessed via selecting the Campaign from the calendar, or via the List Campaign or Show Campaign views.



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1	2 Groundhog Day	3
4	5	6	7	8	9	10
11	12 Lincoln's B-Day	13	14 Valentines Day	15	16	17
18	19 President's Day	20			23	24
25	26	27	28			

January 2007

S	M	T	W	Th	F	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

March 2007

S	M	T	W	Th	F	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

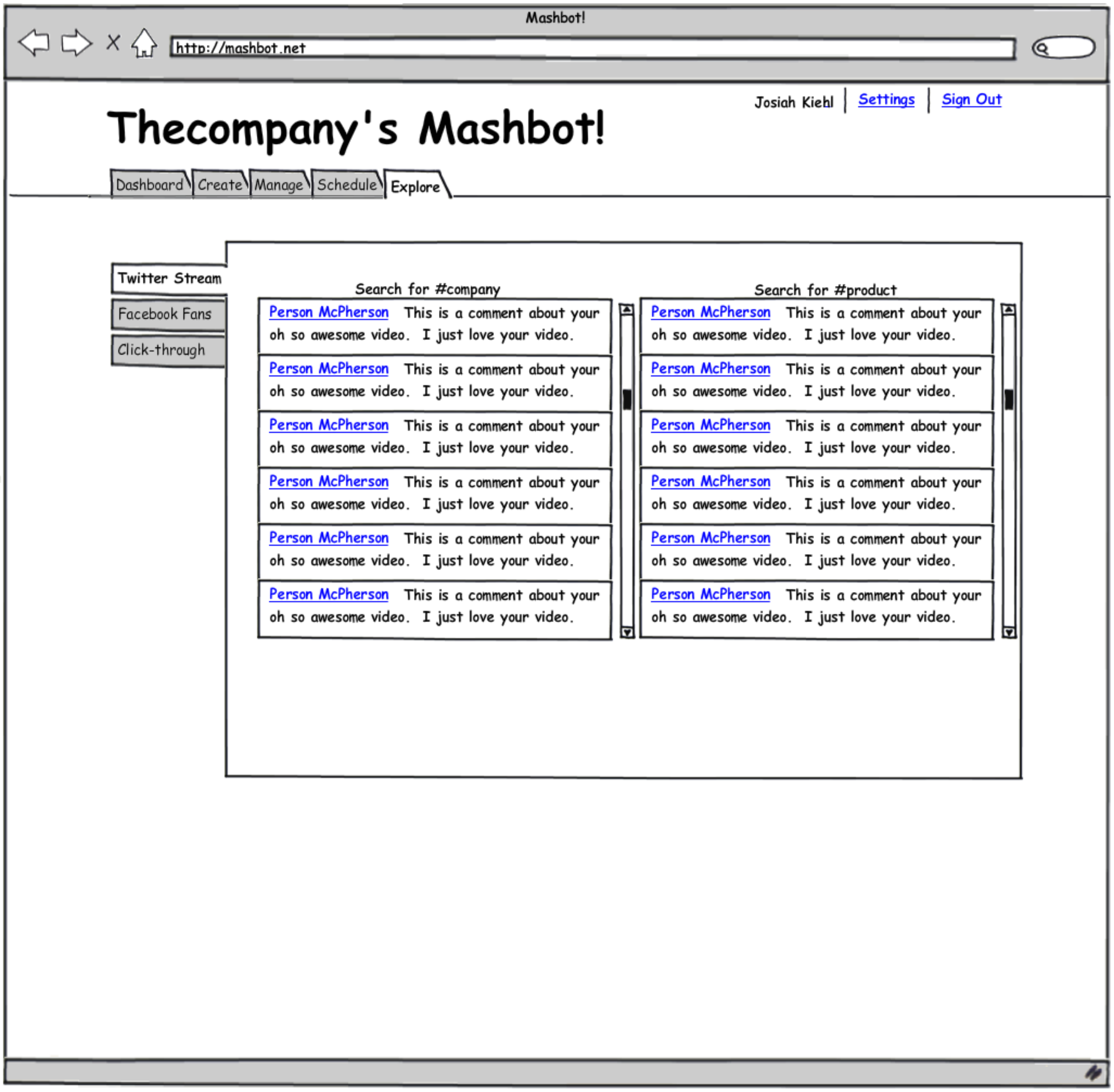
Notes:

www.vertex42.com © 2007 Vertex42 LLC

Content Items can be dragged around the page, each drop setting the new date in the database.

2.5.4 Explore View

Here the user will have several available "Insight Views." These are dependent on which plugins exist in the Publishing and Aggregation Platform, however there will be some provided by the Campaign Manager alone. These will provide charts that are layerable, such that multiple charts can be seen on the same graph.



Thecompany's Mashbot!

Search for #company

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Search for #product

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

Person McPherson This is a comment about your oh so awesome video. I just love your video.

- Plugin Independent
  - Clickthrough tracking — Any time a link is generated via Mashbot, it is given a special redirecting URL that will allow Mashbot to track how many times the link has been clicked.
  - Rate of publishing — How often does the user tweet/blog/etc. This will most likely be used to correlate frequency with user engagement.
- Plugin Dependent
  - Facebook Fan tracking — A line chart of how many fans the user's fan page has.
  - Twitter Follower tracking — A line chart of the number of twitter followers the user's twitter account has.
  - Number of times retweeted — A line chart of the number of times a tweet of the user's has been retweeted.

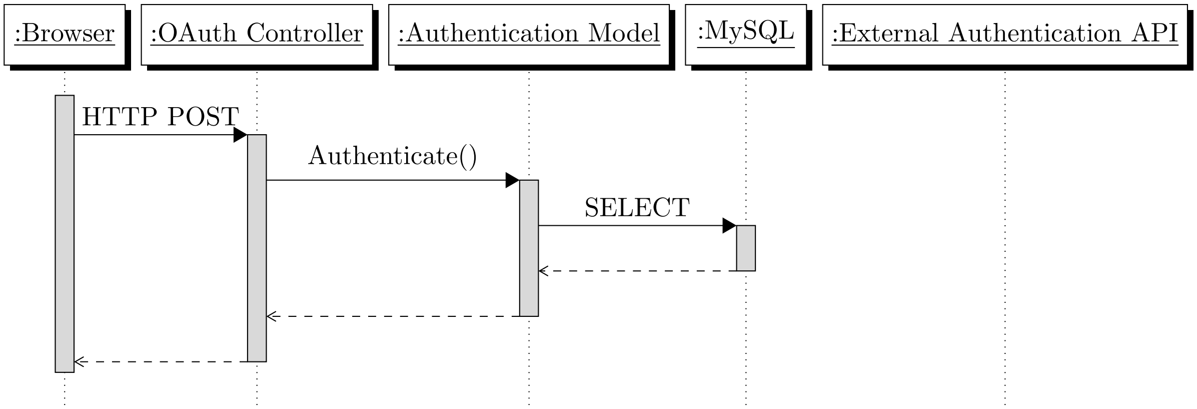


Figure 1: Authentication via OAuth with stored token.

2.6 Business Layer Components

2.7 Session and Authentication

2.7.1 Log In

2.7.2 Log Out

2.7.3 OAuth

2.7.4 Openid

2.7.5 Session Handling

2.8 Data Layer Components

2.9 External Components

2.9.1 Publishing and Aggregation Targets

- Twitter
- Tumblr
- Wordpress
- TODO: etc.

2.9.2 Email/SMTP Service

3 Design Features

3.1 Log In

3.2 External Authentication

3.3 Create User Account

This is a basic CRUD operation: Create User Account.

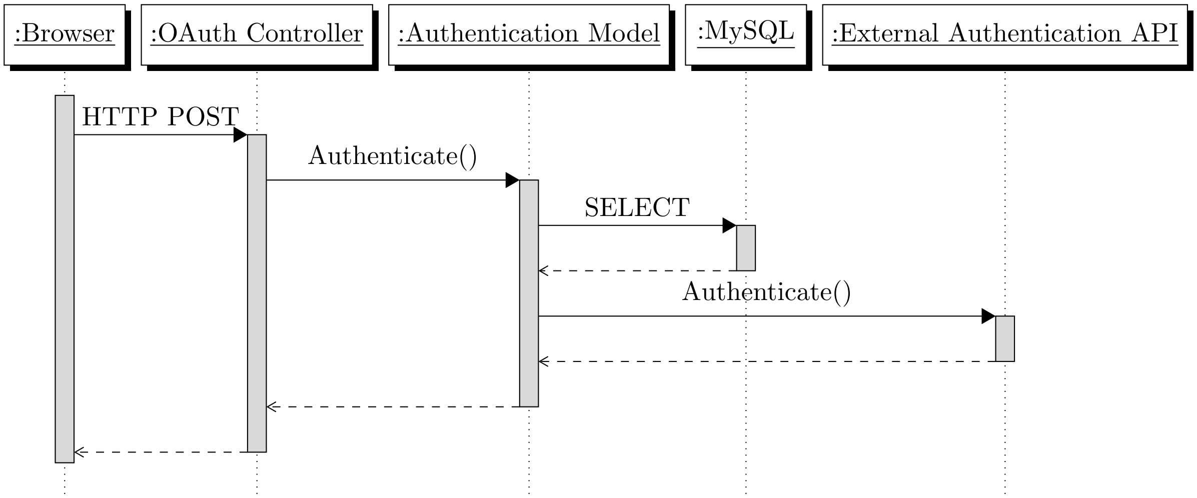


Figure 2: Authentication via OAuth without stored token.

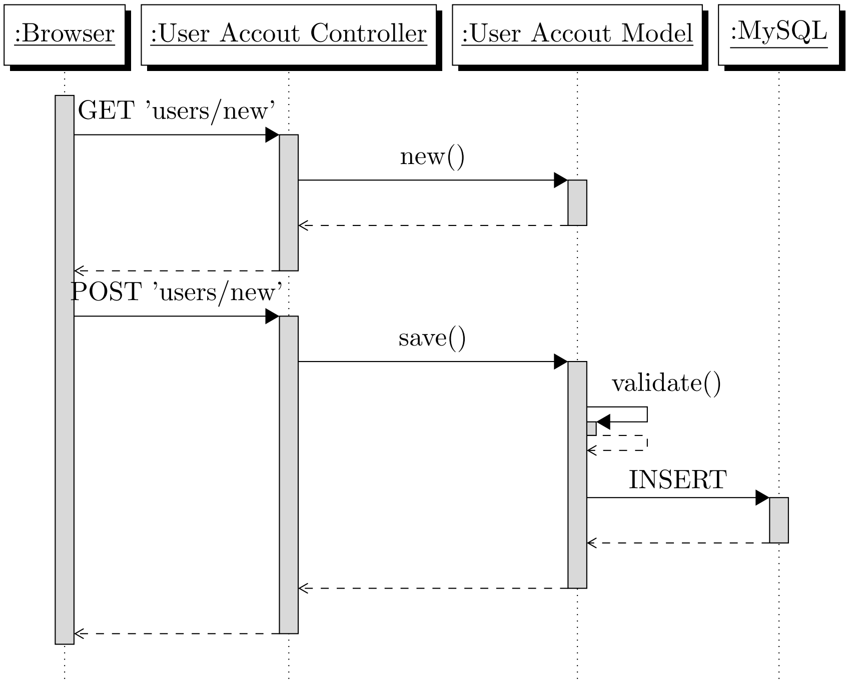


Figure 3: Create User Accout

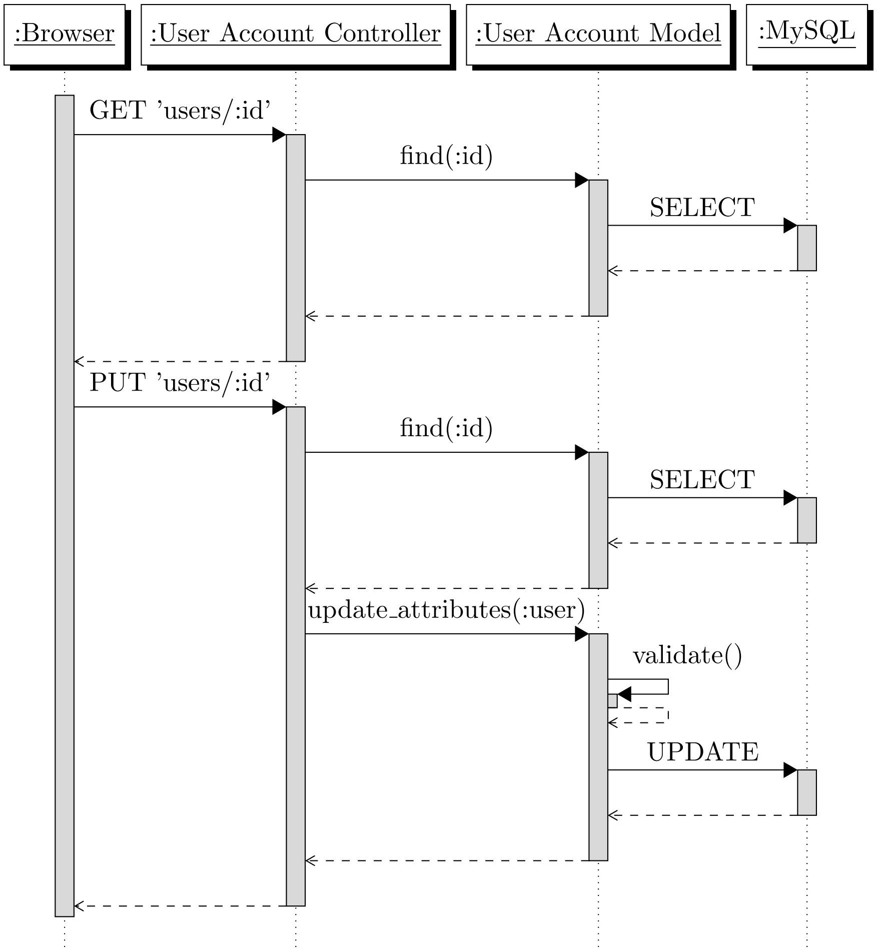


Figure 4: Update a User Account

3.4 Update User Account

This is a basic CRUD operation: Update User Account.

3.5 Delete User Account

This is a basic CRUD operation: Delete User Account.

3.6 Create Campaign

This is a basic CRUD operation: Create Campaign.

3.7 View Campaign

This is a basic CRUD operation: View Campaign.

This is a Read operation, similar to the above, but for one Campaign.

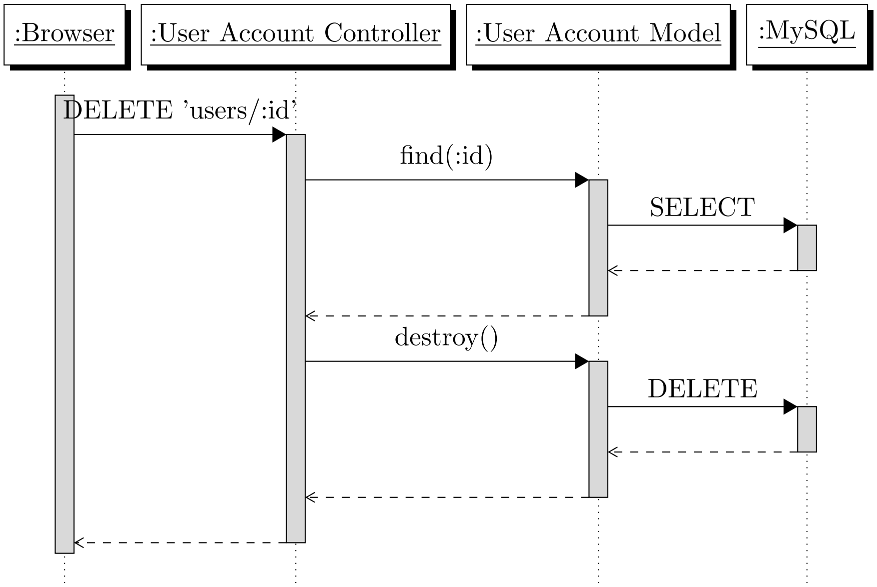


Figure 5: Delete User Account

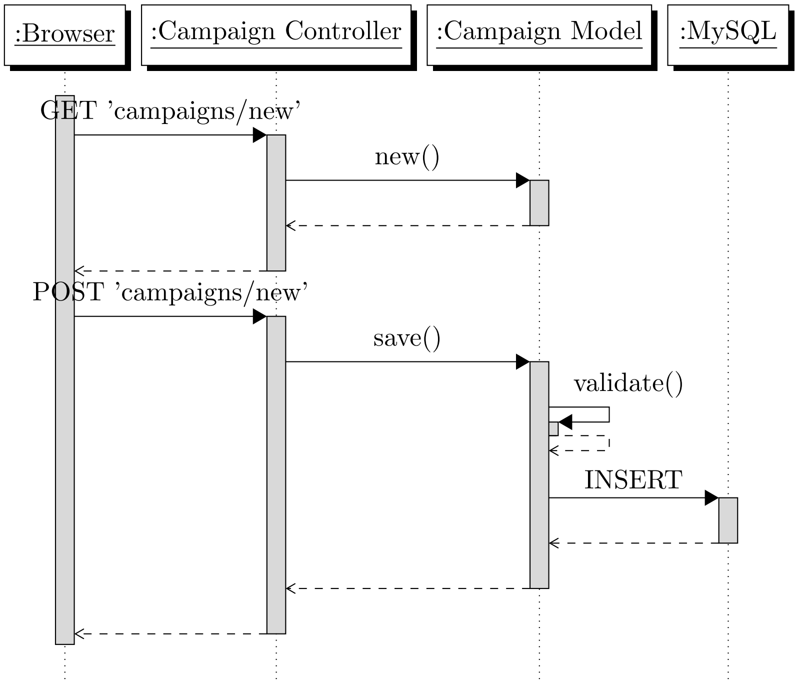


Figure 6: Create Campaign

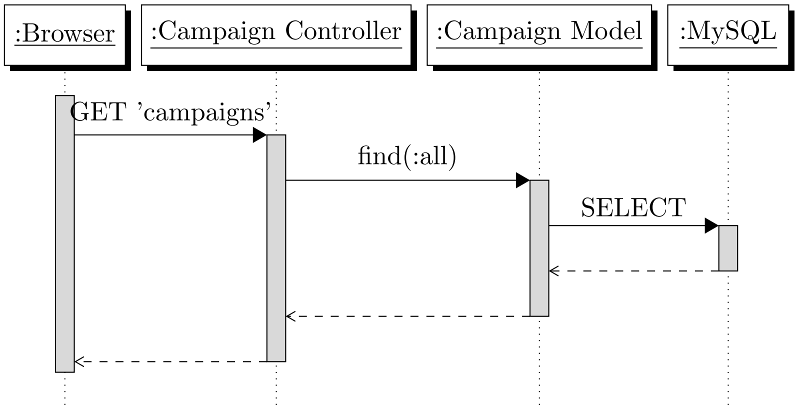


Figure 7: List Campaigns

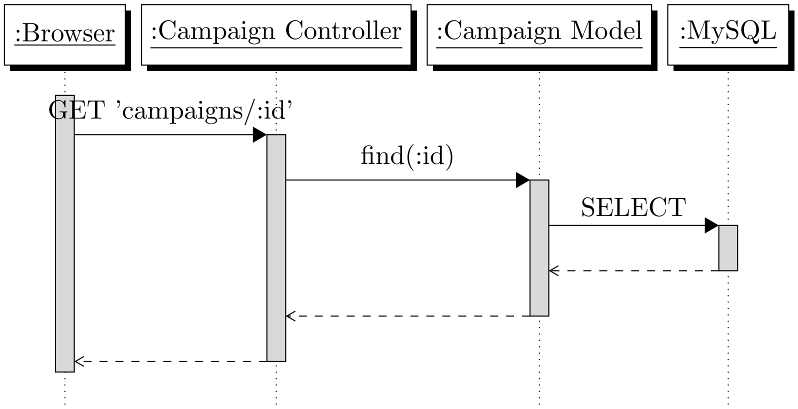


Figure 8: Show a Campaign

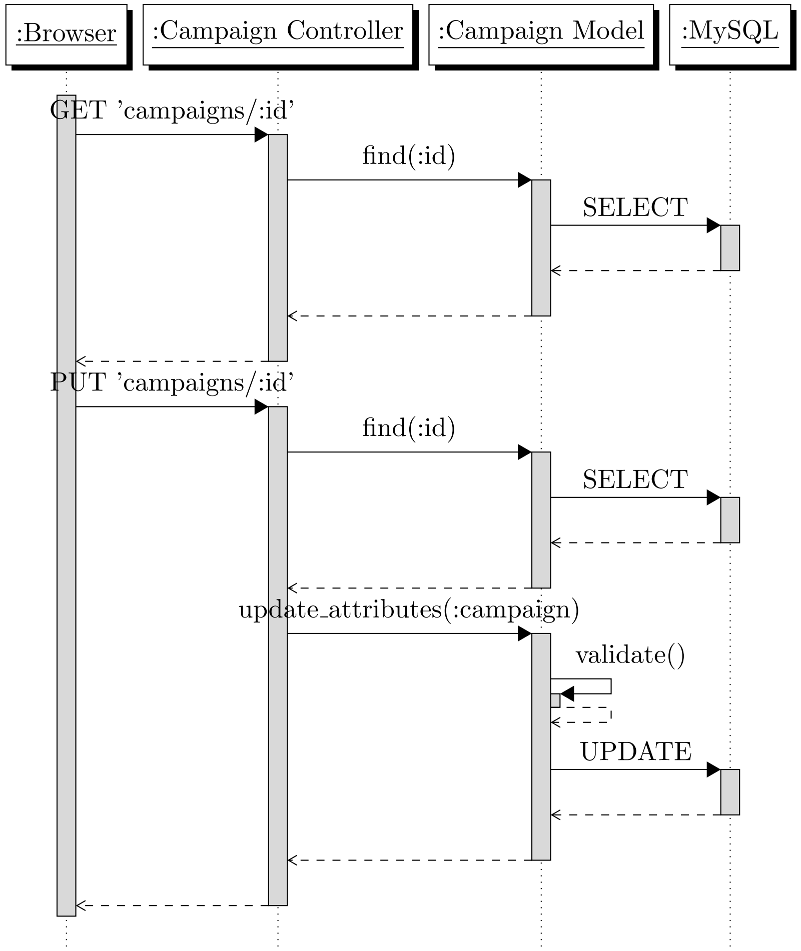


Figure 9: Update a Campaign

3.8 Update Campaign

This is a basic CRUD operation: Update Campaign.

3.9 Delete Campaign

This is a basic CRUD operation: Delete Campaign.

3.10 Schedule Campaign

This is a basic CRUD operation: Schedule Campaign.

3.11 Create Content

This is a basic CRUD operation: Create Content.

3.12 View Content_Unit

This is a basic CRUD operation: View Content_Unit.

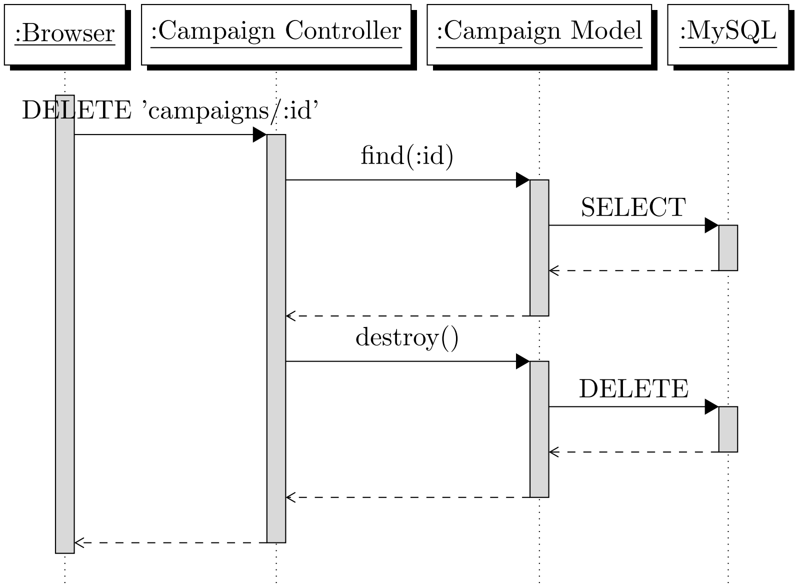


Figure 10: Delete Campaign

This is a Read operation, similar to the above, but for one Content Unit.

3.13 Update Content_Unit

This is a basic CRUD operation: Update Content_Unit.

3.14 Delete Content_Unit

This is a basic CRUD operation: Delete Content_Unit.

3.15 Schedule Content_Unit

In order to change the scheduled time of anything that is schedulable, the same process as an update is carried out.

After content is scheduled, it can be picked up by the Scheduler, triggered via Cron. Every 30 minutes, Cron will run the Scheduler, which will look in the database for active Campaigns. Within those active campaigns, each piece of content will be checked to see if the go-live time is now or past. If it's now or past, the scheduler calls the Publishing and Aggregation Platform with the content needed for a push to the given External Service.

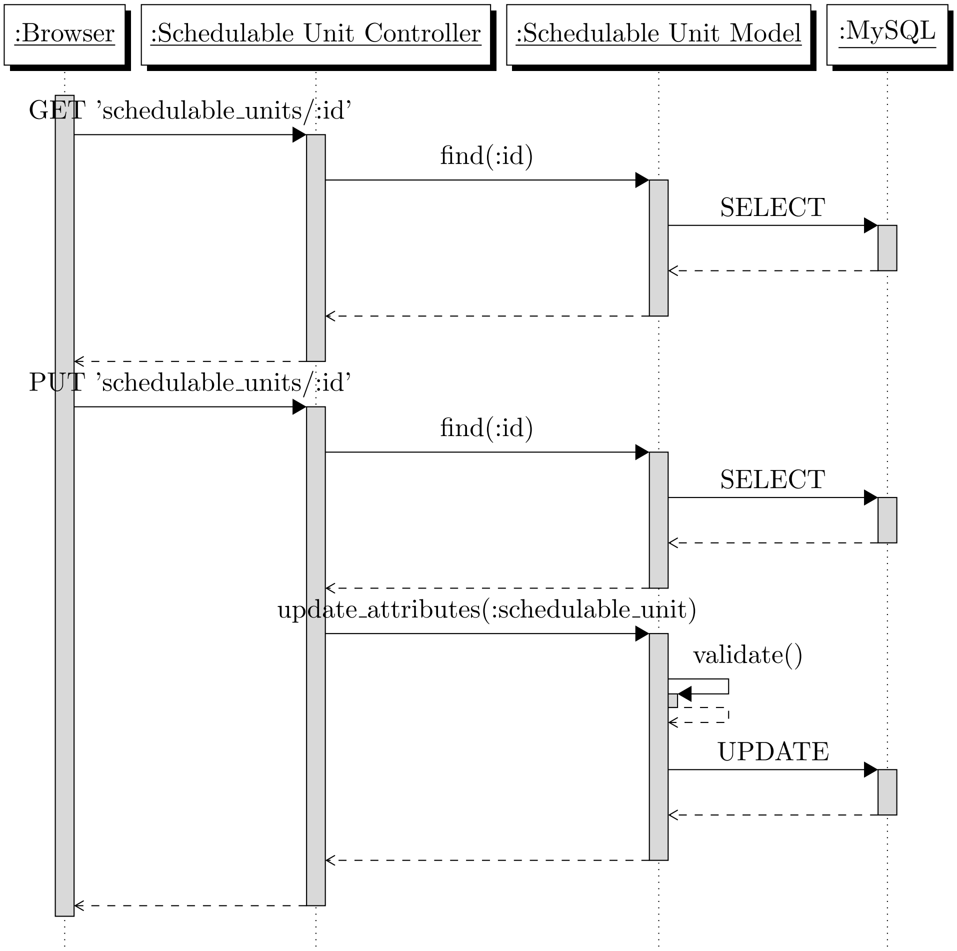

:Browser
:Schedulable Unit Controller
:Schedulable Unit Model
:MySQL
GET 'schedulable_units/:id'
find(:id)
SELECT
PUT 'schedulable_units/:id'
find(:id)
SELECT
update_attributes(:schedulable_unit)
validate()
UPDATE

Figure 11: Update scheduled time of Schdulable Unit

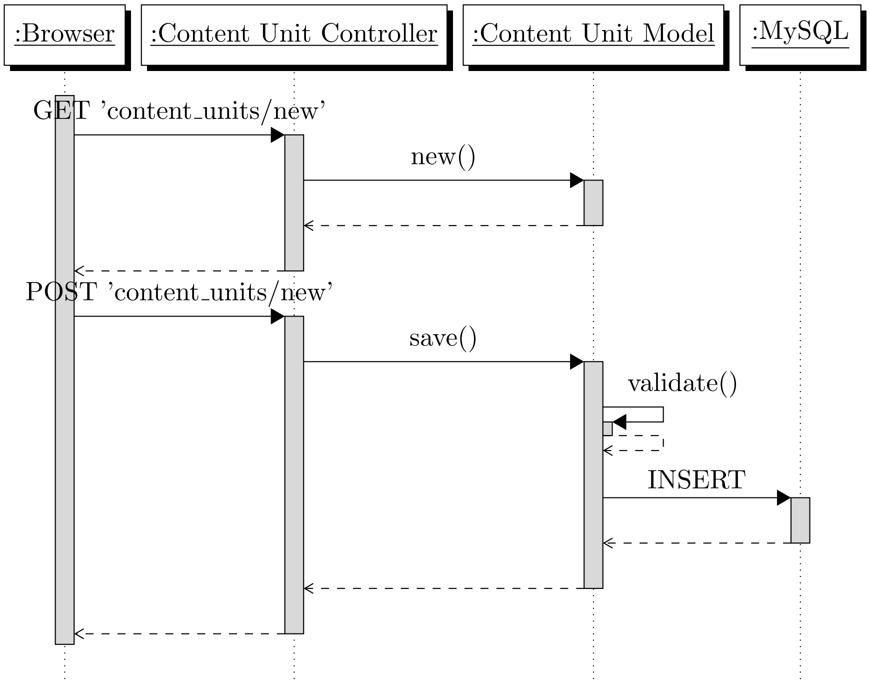


Figure 12: Create Content Unit

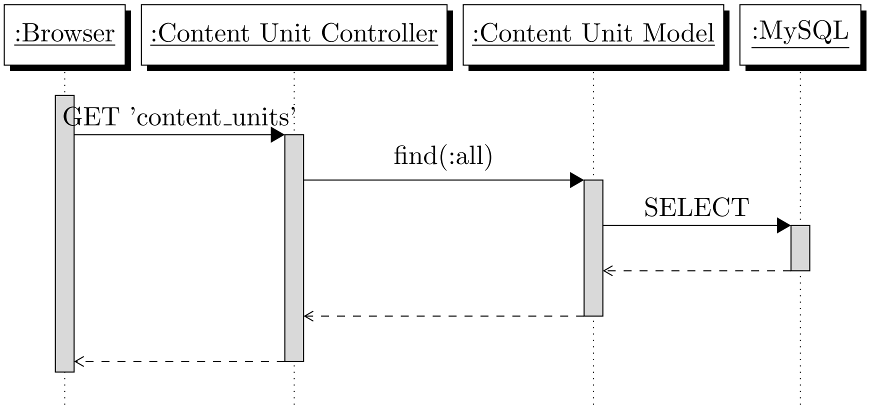


Figure 13: List Content Units

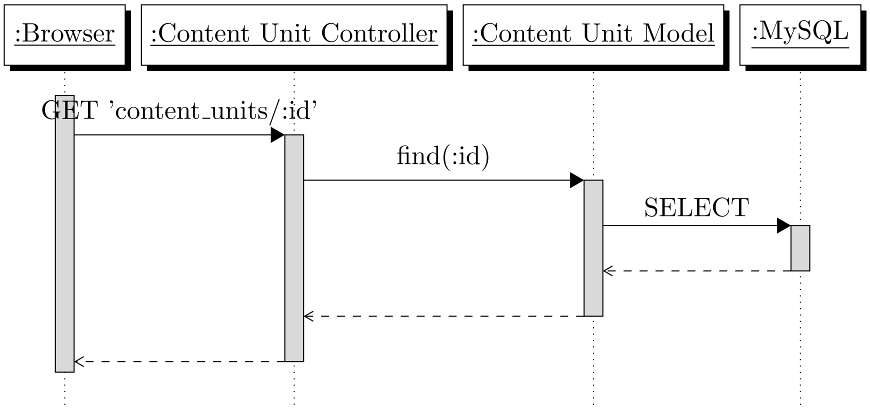


Figure 14: Show a Content Unit

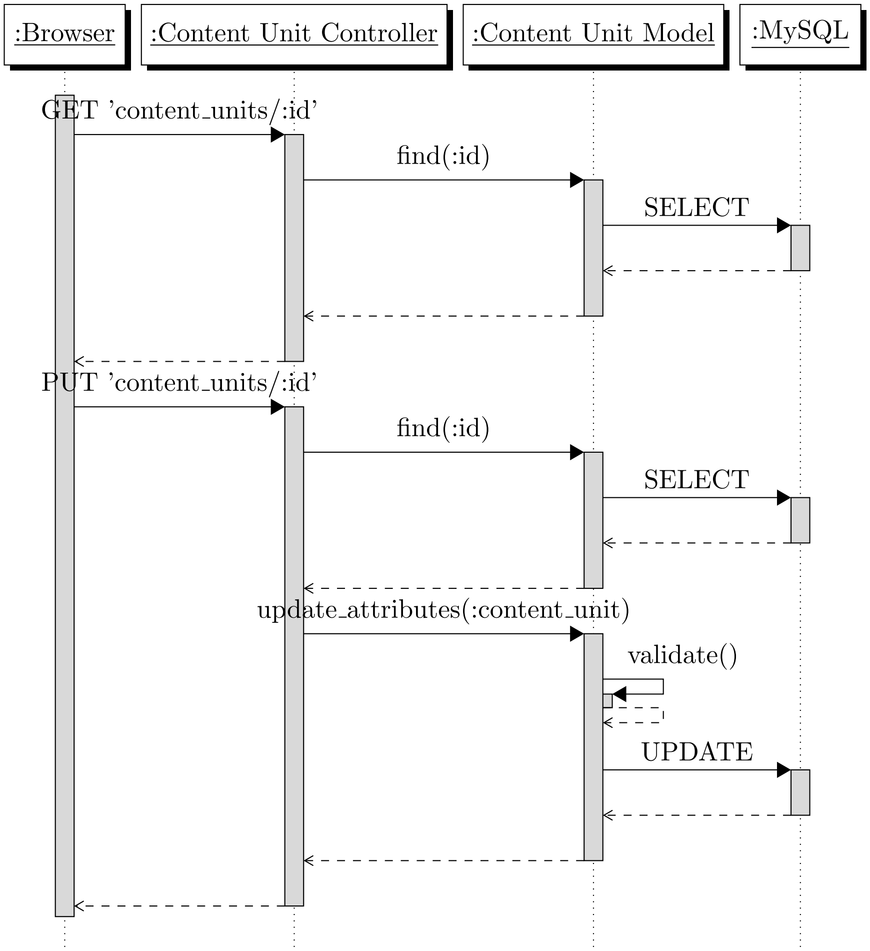


Figure 15: Update content_unit

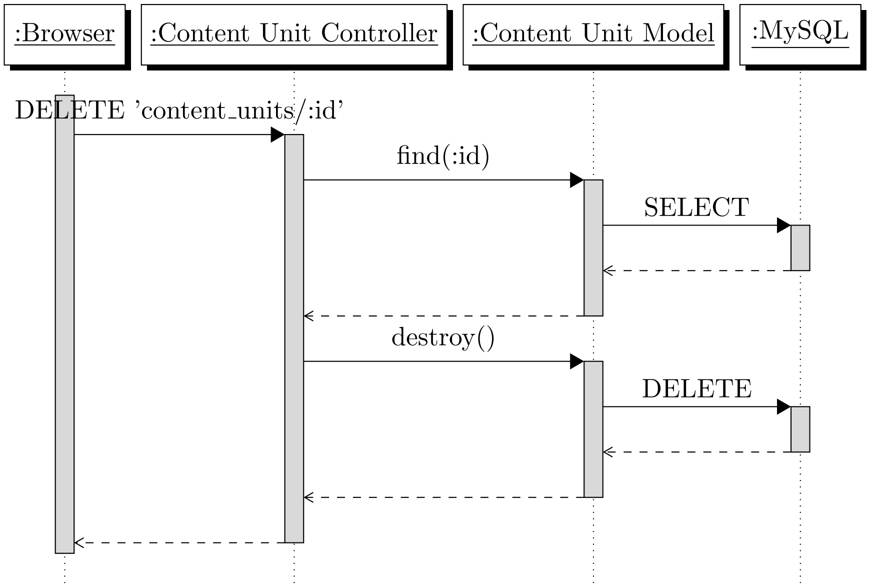


Figure 16: Delete Content Unit

3.16 View Metrics and Statistics via Explore Panel

3.17 Lost User Name

3.18 Lost Password

3.19 Lost Password

4 Database Design

5 Summary

5.1 Advantages of Design

5.2 Disadvantages of Design

5.3 Design Rationale

6 A Requirements Traceability Matrix

6.1 Traceability by Requirement Numbers

6.2 Traceability by Design Component

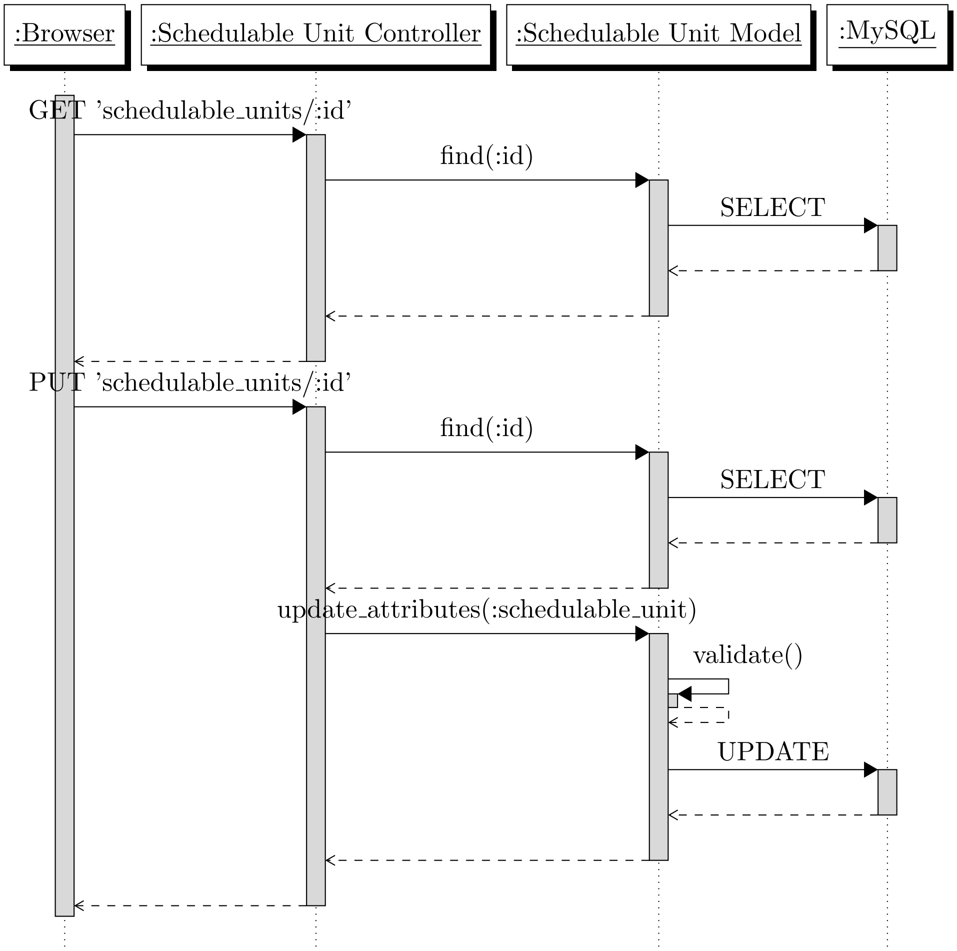

:Browser
:Schedulable Unit Controller
:Schedulable Unit Model
:MySQL
GET 'schedulable_units/:id'
find(:id)
SELECT
PUT 'schedulable_units/:id'
find(:id)
SELECT
update_attributes(:schedulable_unit)
validate()
UPDATE

Figure 17: Update scheduled time of a Schedulable Unit

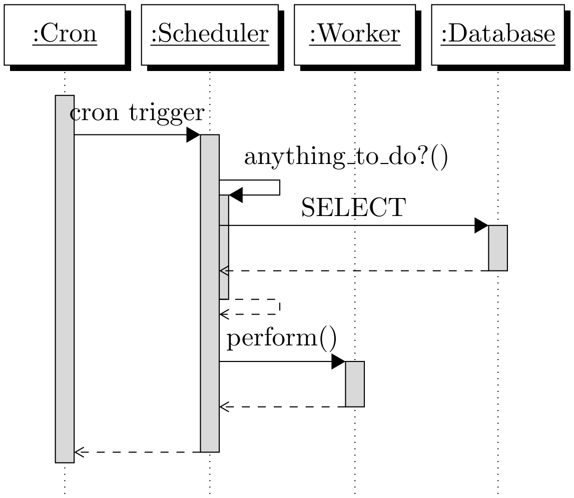


Figure 18: Perform a scheduled action